

Facility Location with Client Latencies: Linear-Programming based Techniques for Minimum-Latency Problems

Deeparnab Chakrabarty*

Chaitanya Swamy†

Abstract

We introduce a problem that is a common generalization of the *uncapacitated facility location* (UFL) and *minimum latency* (ML) problems, where facilities not only need to be opened to serve clients, but also need to be sequentially activated before they can provide service. This abstracts a setting where inventory demanded by customers needs to be stocked or replenished at facilities from a depot or warehouse. Formally, we are given a set \mathcal{F} of n facilities with facility-opening costs $\{f_i\}$, a set \mathcal{D} of m clients, and connection costs $\{c_{ij}\}$ specifying the cost of assigning a client j to a facility i , a root node r denoting the depot, and a *time metric* d on $\mathcal{F} \cup \{r\}$. Our goal is to open a subset F of facilities, find a path P starting at r and spanning F to activate the open facilities, and connecting each client j to a facility $\phi(j) \in F$, so as to minimize $\sum_{i \in F} f_i + \sum_{j \in \mathcal{D}} (c_{\phi(j),j} + t_j)$, where t_j is the time taken to reach $\phi(j)$ along path P . We call this the *minimum latency uncapacitated facility location* (MLUFL) problem.

Our main result is an $O(\log n \max\{\log n, \log m\})$ -approximation for MLUFL. Via a reduction to the group Steiner tree (GST) problem, we show this result is tight in the sense that any improvement in the approximation guarantee for MLUFL, implies an improvement in the (currently known) approximation factor for GST. We obtain significantly improved constant approximation guarantees for two natural special cases of the problem: (a) *related MLUFL*, where the connection costs form a metric that is a scalar multiple of the time metric; (b) *metric uniform MLUFL*, where we have metric connection costs and the time-metric is uniform. Our LP-based methods are fairly versatile and are easily adapted with minor changes to yield approximation guarantees for MLUFL (and ML) in various more general settings, such as (i) the setting where the latency-cost of a client is a function (of bounded growth) of the delay faced by the facility to which it is connected; and (ii) the k -route version, where we can dispatch k vehicles in parallel to activate the open facilities.

Our LP-based understanding of MLUFL also offers some LP-based insights into ML. We obtain two natural LP-relaxations for ML with constant integrality gap, which we believe shed new light upon the problem and offer a promising direction for obtaining improvements for ML.

1 Introduction

Facility location and vehicle routing problems are two broad classes of combinatorial optimization problems that have been widely studied in the Operations Research community (see, e.g., [25, 32]), and have a wide range of applications. Both problems can be described in terms of an underlying set of clients that need to be serviced. In facility location problems, there is a candidate set of facilities that provide service, and the goal is to open some facilities and connect each client to an open facility so as to minimize some combination of the facility-opening and client-connection costs. Vehicle routing problems consider the setting where a vehicle (delivery-man or repairman) provides service, and the goal is to plan a route that visits (and hence services) the clients as quickly as possible. Two common objectives considered are: (i) minimize the total

*deepc@seas.upenn.edu. Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA 19104. Most of the work was done as a postdoctoral fellow at the Dept. of Combinatorics and Optimization, Univ. Waterloo.

†cswamy@math.uwaterloo.ca. Dept. of Combinatorics and Optimization, Univ. Waterloo, Waterloo, ON N2L 3G1. Supported in part by NSERC grant 327620-09 and an Ontario Early Researcher Award.

length of the vehicle’s route, giving rise to the traveling salesman problem (TSP), and (ii) (adopting a client-oriented approach) minimize the sum of the client delays, giving rise to minimum latency (ML) problems.

These two classes of problems have mostly been considered separately. However, various logistics problems involve both facility-location and vehicle-routing components. For example, consider the following oft-cited prototypical example of a facility location problem: a company wants to determine where to open its retail outlets so as to serve its customers effectively. Now, inventory at the outlets needs to be replenished or ordered (e.g., from a depot); naturally, a customer cannot be served by an outlet unless the outlet has the inventory demanded by it, and delays incurred in procuring inventory might adversely impact customers. Hence, it makes sense for the company to *also* keep in mind the latencies faced by the customers while making its decisions about where to open outlets, how to connect customers to open outlets, and in what order to replenish the open outlets, thereby adding a vehicle-routing component to the problem.

We propose a mathematical model that is a common generalization of the *uncapacitated facility location* (UFL) and *minimum latency* (ML) problems, and abstracts a setting (such as above) where facilities need to be “activated” before they can provide service. Formally, as in UFL, we have a set \mathcal{F} of n facilities, and a set \mathcal{D} of m clients. Opening facility i incurs a *facility-opening cost* f_i , and assigning a client j to a facility i incurs *connection cost* c_{ij} . Taking a lead from minimum latency problems, we model activation delays as follows. We have a root (depot) node r , and a *time metric* d on $\mathcal{F} \cup \{r\}$. A feasible solution specifies a subset $F \subseteq \mathcal{F}$ of facilities to open, a path P starting at r and spanning F along which the open facilities are activated, and assigns each client j to an open facility $\phi(j) \in F$. The cost of such a solution is

$$\sum_{i \in F} f_i + \sum_{j \in \mathcal{D}} (c_{\phi(j)j} + t_j) \quad (1)$$

where $t_j = d_P(r, \phi(j))$ is the time taken to reach facility $\phi(j)$ along path P . We refer to t_j as client j ’s latency cost. The goal is to find a solution with minimum total cost. We call this the *minimum-latency uncapacitated facility location* (MLUFL) problem.

Apart from being a natural problem of interest, we find MLUFL appealing since it generalizes, or is closely-related to, various diverse problems of interest (in addition to UFL and ML); our work yields new insights on some of these problems, most notably ML (see “Our results”). One such problem, which captures much of the combinatorial core of MLUFL is what we call the *minimum group latency* (MGL) problem. Here, we are given an undirected graph with metric edge weights $\{d_e\}$, subsets $\{G_j\}$ of vertices called groups, and a root r ; the goal is to find a path starting at r that minimizes the sum of the cover times of the groups, where the cover time of G_j is the first time at which some $i \in G_j$ is visited on the path. Observe that MGL can be cast as MLUFL with zero facility costs (where $\mathcal{F} = \text{node-set} \setminus \{r\}$), where for each group G_j , we create a client j with $c_{ij} = 0$ if $i \in G_j$ and ∞ otherwise. Note that we may assume that the groups are disjoint (by creating multiple co-located copies of a node), in which case these c_{ij} s form a metric. MGL itself captures various other problems. Clearly, when each G_j is a singleton, we obtain the minimum latency problem. Also, given a set-cover instance, if we consider a graph whose nodes are $(r$ and) the sets, create a group G_j for each element j consisting of the sets containing it, and consider the uniform metric, then this MGL problem is simply the *min-sum set cover* (MSSC) problem [16].

Our results and techniques. Our main result is an $O(\log n \max\{\log m, \log n\})$ -approximation algorithm for MLUFL (Section 2.1), which for the special case of MGL, implies an $O(\log^2 n)$ approximation. Complementing this result, we prove (Theorem 2.9) that a ρ -approximation algorithm (even) for MGL yields an $O(\rho \log m)$ -approximation algorithm for the *group Steiner tree* (GST) problem [17] on n nodes and m groups. Thus, any improvement in our approximation ratio for MLUFL would yield a corresponding improvement of GST, whose approximation ratio has remained at $O(\log^2 n \log m)$ for a decade [17]. Moreover, combined with the result of [22] on the inapproximability of GST, this shows that MGL, and hence MLUFL with metric connection costs cannot be approximated to better than a $\Omega(\log m)$ -factor unless $NP \subseteq ZTIME(n^{\text{polylog}(n)})$.

Given the above hardness result, we investigate certain well-motivated special cases of MLUFL and obtain significantly improved performance guarantees. In Section 2.2, we consider the case where the connection costs form a metric, which is a scalar multiple of the d -metric (i.e., $d_{uv} = c_{uv}/M$, where $M \geq 1$; the problem is trivial if $M < 1$). For example, in a supply-chain logistics problem, this models a natural setting where the connection of clients to facilities, and the activation of facilities both proceed along the same transportation network. We obtain a *constant-factor* approximation algorithm for this problem.

In Section 2.3, we consider the *uniform MLUFL* problem, which is the special case where the time-metric is uniform. Uniform MLUFL already generalizes MSSC (and also UFL). For uniform MLUFL with metric connection costs (i.e., metric uniform MLUFL), we devise a 10.78-approximation algorithm. (Without metricity, the problem becomes set-cover hard, and we obtain a simple matching $O(\log m)$ -approximation.) The chief novelty here lies in the technique used to obtain this result. We give a simple generic reduction (Theorem 2.12) that shows how to reduce the metric uniform MLUFL problem with facility costs to one *without* facility costs, in conjunction with an algorithm for UFL. This reduction is surprisingly robust and versatile and has other applications. For example, the same reduction yields an $O(1)$ -approximation for *metric uniform k -median* (i.e., metric uniform MLUFL where at most k facilities may be opened), and the same ideas lead to improved guarantees for the k -median versions of connected facility location [31], and facility location with service installation costs [28].

We obtain our approximation bounds by rounding the optimal solution to a suitable linear-programming (LP) relaxation of the problem. This is interesting since we are not aware of any previous LP-based methods to attack ML (as a whole). In Section 3, we leverage this to obtain some interesting insights about ML, which we believe cast new light on the problem. In particular, we present two LP-relaxations for ML, and prove that these have (small) constant integrality gap. Our first LP is a specialization of our LP-relaxation for MLUFL. Interestingly, the integrality-gap bound for this LP relies only on the fact that the natural LP relaxation for TSP has constant integrality gap (i.e., a ρ -integrality gap for the natural TSP LP relaxation translates to an $O(\rho)$ -integrality gap). In contrast, the various known algorithms for ML [7, 10, 1] all utilize algorithms for the arguably harder k -MST problem or its variants. Our second LP has exponentially-many variables, one for every path (or tree) of a given length bound, and the separation oracle for the dual problem is a rooted path (or tree) orienteering problem: given rewards on the nodes and metric edge costs, find a (simple) path rooted at r of length at most B that gathers maximum reward. We prove that even a bicriteria approximation for the orienteering problem yields an approximation for ML while losing a constant factor. This connection between orienteering and ML is known [14]. But we feel that our alternate proof, where the orienteering problem appears as the separation oracle required to solve the dual LP, offers a more illuminating explanation of the relation between the approximability of the two problems. (In fact, the same relationship also holds between MGL and “group orienteering.”)

We believe that the use of LPs opens up ML to new venues of attack. A good LP-relaxation is beneficial because it yields a concrete, tractable lower bound and handle on the integer optimum, which one can exploit to design algorithms (a point repeatedly driven home in the field of approximation algorithms). Also, LP-based techniques tend to be fairly versatile and can be adapted to handle more general variants of the problem (more on this below). Our LP-rounding algorithms exploit various ideas developed for scheduling and facility-location problems (e.g., α -points) and polyhedral insights for TSP, which suggests that the wealth of LP-based machinery developed for these problems can be leveraged to obtain improvements for ML. We suspect that our LP-relaxations are in fact better than what we have accounted for, and consider them to be a promising direction for making progress on ML.

Section 4 showcases the flexibility afforded by our LP-based techniques, by showing that our algorithms and analyses extend with little effort to handle various generalizations of MLUFL (and hence, ML). For example, consider the setting where the latency-cost of a client j is $\lambda(\text{time taken to reach the facility serving } j)$, for some non-decreasing function $\lambda(\cdot)$. When λ is convex and has “growth” at most p (i.e., $\lambda(cx) \leq c^p \lambda(x)$), we derive an $O(\max\{(p \log^2 n)^p, p \log n \log m\})$ -approximation for MLUFL, an $O(2^{O(p)})$ -approximation

for related MLUFL and ML, and an $O(1)$ -approximation for metric uniform MLUFL. (Concave λ s are even easier to handle.) This in turn leads to approximation guarantees for the \mathcal{L}_p -norm generalization of MLUFL, where instead of the sum (i.e., \mathcal{L}_1 -norm) of client latencies, the \mathcal{L}_p -norm of the client latencies appears in the objective function. The spectrum of \mathcal{L}_p norms tradeoff efficiency with fairness, making the \mathcal{L}_p -norm problem an appealing problem to consider. We obtain an $O(p \log n \max\{\log n, \log m\})$ -approximation for MLUFL, and an $O(1)$ -approximation for the other special cases. Another notable extension is the k -route version of the problem, where we may use k paths starting at r to traverse the open facilities. With one simple modification to our LPs (and algorithms), *all* our approximation guarantees translate to this setting. As a corollary, we obtain a constant-factor approximation for the \mathcal{L}_p -norm version of the k -traveling repairmen problem [14] (the k -route version of ML).

Related work. To the best of our knowledge, MLUFL and MGL are new problems that have not been studied previously. There is a great deal of literature on facility location and vehicle routing problems (see, e.g., [25, 32]) in general, and UFL and ML, in particular, which are special cases of our problem, and we limit ourselves to a sampling of some of the relevant results. The first constant approximation guarantee for UFL was obtained by Shmoys, Tardos, and Aardal [29] via an LP-rounding algorithm, and the current state-of-the-art is 1.5-approximation algorithm due to Byrka [8]. The minimum latency (ML) problem seems to have been first introduced to the computer science community by Blum et al. [7], who gave a constant-factor approximation algorithm for it. Goemans and Kleinberg [19] improved the approximation factor to 10.78, using a “tour-concatenation” lemma, which has formed a component of all subsequent algorithms and improvements for ML. The current-best approximation factor for ML is 3.59 due to Chaudhuri, Godfrey, Rao and Talwar [10]. As mentioned earlier, MGL with a uniform time metric captures the min-sum set cover (MSSC) problem. This problem was introduced by Feige, Lovasz and Tetali [16], who gave a 4-approximation algorithm for the problem and a matching inapproximability result. Recently, Azar et al. [2] introduced a generalization of MSSC, for which Bansal et al. [5] obtained a constant-factor approximation.

If instead of adding up the latency cost of clients, we include the *maximum* latency cost of a client in the objective function of MLUFL, then we obtain the min-max versions of MGL and MLUFL, which have been studied previously. The min-max version of MGL is equivalent to a “path-variant” of GST: we seek a path starting at r of minimum total length that covers every group. Garg, Konjevod, and Ravi [17] devised an LP-rounding based $O(\log^2 n \log m)$ -approximation for GST, where n is the number of nodes and m is the number of groups. Charikar et al. [9] gave a deterministic algorithm with the same guarantee, and Halperin and Krauthgamer [22] proved an $\Omega(\log^2 m)$ -inapproximability result. The rounding technique of [17] and the deterministic tree-embedding construction of [9] (rather its improvement by [15]) are key ingredients of our algorithm for (general) MLUFL. The min-max version of MLUFL can be viewed a path-variant of connected facility location [27, 20]. The connected facility location problem, in its full generality, is essentially equivalent to GST [27]; however, if the connection costs form a metric, and the time- and the connection-cost metrics are scalar multiples of each other, then various constant-factor approximations are known [20, 31, 13].

Very recently, we have learnt that, independent of, and concurrent with, our work, Gupta et al. [21] also propose the minimum group latency (MGL) problem (which they arrive at in the course of solving a different problem), and obtain results similar to ours for MGL. They also obtain an $O(\log^2 n)$ -approximation for MGL, and the reduction from GST to MGL with a $\log m$ -factor loss (see also [26]), and relate the approximability of the MGL and “group orienteering” problems. Their techniques are combinatorial and not LP-based, and it is not clear how these can be extended to handle facility-opening costs.

2 LP-rounding approximation algorithms for MLUFL

We can express MLUFL as an integer program and relax the integrality constraints to obtain a linear program as follows. We may assume that $d_{ii'}$ is integral for all $i, i' \in \mathcal{F} \cup \{r\}$. Let E denote the edge-set of the complete graph on $\mathcal{F} \cup \{r\}$ and let $d_{\max} := \max_{e \in E} d_e$. Let $T \leq \min\{n, m\}d_{\max}$ be a known upper bound on the maximum activation time of an open facility in an optimal solution. For every facility i , client j , and time $t \leq T$, we have a variable $y_{i,t}$ indicating if facility i is opened at time t or not, and a variable $x_{ij,t}$ indicating whether client j connects to facility i at time t . Also, for every edge $e \in E$ and time t , we introduce a variable $z_{e,t}$ which denotes if edge e has been traversed by time t . Throughout, we use i to index the facilities in \mathcal{F} , j to index the clients in \mathcal{D} , t to index the time units in $[T] := \{1, \dots, T\}$, and e to index the edges in E .

$$\min \quad \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} (c_{ij} + t) x_{ij,t} \quad (\text{P})$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i,t} x_{ij,t} \geq 1 && \text{for all } j; && x_{ij,t} \leq y_{i,t} && \text{for all } i, j, t \\ & \sum_e d_e z_{e,t} \leq t && \text{for all } t \end{aligned} \quad (2)$$

$$\sum_{e \in \delta(S)} z_{e,t} \geq \sum_{i \in S, t' \leq t} x_{ij,t'} \quad \text{for all } t, S \subseteq \mathcal{F}, j \quad (3)$$

$$x_{ij,t}, y_{i,t}, z_{e,t} \geq 0 \quad \text{for all } i, j, t, e; \quad y_{i,t} = 0 \quad \text{for all } i, t \text{ with } d_{ir} > t.$$

The first two constraints encode that each client is connected to some facility at some time, and that if a client is connected to a facility i at time t , then i must be open at time t . Constraint (2) ensures that by time t no more than t “distance” is covered by the tour on facilities, and (3) ensures that if a client is connected to i by time t , then the tour must have visited i by time t . We assume for now that $T = \text{poly}(m)$, and show later how to remove this assumption (Lemma 2.7, Theorem 2.8). Thus, (P) can be solved efficiently since one can efficiently separate over the constraints (3). Let (x, y, z) be an optimal solution to (P), and OPT denote its objective value. For a client j , define $C_j^* = \sum_{i,t} c_{ij} x_{ij,t}$, and $L_j^* = \sum_{i,t} t x_{ij,t}$. We devise various approximation algorithms for MLUFL by rounding (x, y, z) to an integer solution.

In Section 2.1, we give a polylogarithmic approximation algorithm for (general) MLUFL (where the c_{ij} s need not even form a metric). Complementing this result, we prove (Theorem 2.9) that a ρ -approximation algorithm (not necessarily LP-based) for MLUFL yields an $O(\rho \log m)$ -approximation algorithm for the GST problem on n nodes and m groups. In Sections 2.2 and 2.3, we obtain significantly-improved approximation guarantees for various well-motivated special cases of MLUFL. Section 2.2 obtains a *constant-factor* approximation algorithm in the natural setting where the connection costs form a metric that is a scalar multiple of the time-metric. Section 2.3 considers the setting where the time-metric is the uniform metric. Our main result here is a constant-factor approximation for metric connection costs, which is obtained via a rather versatile reduction of this uniform MLUFL problem to UFL and uniform MLUFL with zero-facility costs.

2.1 An $O(\log n \cdot \max\{\log n, \log m\})$ -approximation algorithm

We first give an overview. Let $N_j = \{i \in \mathcal{F} : c_{ij} \leq 4C_j^*\}$ be the set of facilities “close” to j , and define τ_j as the earliest time t such that $\sum_{i \in N_j, t' \leq t} x_{ij,t'} \geq \frac{2}{3}$. By Markov’s inequality, we have $\sum_{i \in N_j} \sum_t x_{ij,t} \geq \frac{2}{4}$ and $\tau_j \leq 12L_j^*$. It is easiest to describe the algorithm assuming first that the time-metric d is a tree metric. Our algorithm runs in phases, with phase ℓ corresponding to time $t_\ell = 2^\ell$. In each phase, we compute a random subtree rooted at r of “low” cost such that for every client j with $\tau_j \leq t_\ell$, with constant probability, this tree contains a facility in N_j . To compute this tree, we utilize the rounding procedure of Garg-Konjevod-Ravi (GKR) for the *group Steiner tree* (GST) problem [17] (see Lemma 2.4), by creating a group for each

client j with $\tau_j \leq t_\ell$ comprising of, roughly speaking, the facilities in N_j . We open all the facilities included in the subtree, and obtain a tour via the standard trick of doubling all edges and performing an Eulerian tour with possible shortcutting. The overall tour is a concatenation of all the tours obtained in the various phases. For each client j , we consider the first tree that contains a facility from N_j (which must therefore be open), and connect j to such a facility.

Given the result for tree metrics, an oft-used idea to handle the case when d is not a tree metric is to approximate it by a distribution of tree metrics with $O(\log n)$ distortion [15]. Our use of this idea is however more subtle than the typical applications of probabilistic tree embeddings. Instead of moving to a distribution over tree metrics up front, in each phase ℓ , we use the results of [9, 15] to *deterministically* obtain a tree \mathcal{T}_ℓ with edge weights $\{d_{\mathcal{T}_\ell}(e)\}$, such that the resulting tree metric dominates d and $\sum_{e=(i,i')} d_{\mathcal{T}_\ell}(i,i') z_{e,t_\ell} = O(\log n) \sum_e d_e z_{e,t_\ell}$. As we show in Section 4, this deterministic choice allows to extend our algorithm and analysis effortlessly to the setting where the latency-cost in the objective function is measured by a more general function (e.g., the \mathcal{L}_p -norm) of the client-latencies. The algorithm is described in detail as Algorithm 1, and utilizes the following results. Let $\tau_{\max} = \max_j \tau_j$.

Theorem 2.1 ([9, 15]) *Given any edge weights $\{\mathfrak{z}_e\}_{e \in E}$, one can deterministically construct a weighted tree \mathcal{T} having leaf-set $\mathcal{F} \cup \{r\}$, leading to a tree metric, $d_{\mathcal{T}}(\cdot)$, such that, for any $i, i' \in \mathcal{F} \cup \{r\}$, we have: (i) $d_{\mathcal{T}}(i, i') \geq d_{i,i'}$, and (ii) $\sum_{e=(i,i') \in E} d_{\mathcal{T}}(i, i') \mathfrak{z}_{i,i'} = O(\log n) \sum_e d_e \mathfrak{z}_e$.*

Theorem 2.2 ([17]) *Consider a tree \mathcal{T} rooted at r with n leaves, subsets G_1, \dots, G_p of leaves, and fractional values \mathfrak{z}_e on the edges of \mathcal{T} satisfying $\mathfrak{z}(\delta(S)) \geq \nu_j$ for every group G_j and node-set S such that $G_j \subseteq S$, where $\nu_j \in [\frac{1}{2}, 1]$. There exists a randomized polytime algorithm, henceforth called the GKR algorithm, that returns a rooted subtree $T'' \subseteq \mathcal{T}$ such that (i) $\Pr[e \in T''] \leq \mathfrak{z}_e$ for every edge $e \in \mathcal{T}$; and (ii) $\Pr[T'' \cap G_j = \emptyset] \leq \exp(-\frac{\nu_j}{64 \log_2 n})$ for every group G_j .*

Algorithm 1 Given: a fractional solution (x, y, z) to (P) (with C_j^*, L_j^*, N_j , and τ_j defined as above for each client j).

- A1. In each phase $\ell = 0, 1, \dots, \mathcal{N} := \lceil \log_2(2\tau_{\max}) + 4 \log_2 m \rceil$, we do the following. Let $t_\ell = \min\{2^\ell, T\}$.
 - A1.1. Use Theorem 2.1 with edge weights $\{z_{e,t_\ell}\}$ to obtain a tree $\mathcal{T}_\ell = (V(\mathcal{T}_\ell), E(\mathcal{T}_\ell))$. Extend \mathcal{T}_ℓ to a tree \mathcal{T}'_ℓ by adding a dummy leaf edge (i, v_i) of cost f_i to \mathcal{T}_ℓ for each facility i . Let $E' = \{(i, v_i) : i \in \mathcal{F}\}$.
 - A1.2. Map the LP-assignment $\{z_{e,t_\ell}\}_{e \in E}$ to an assignment \mathfrak{z} on the edges of \mathcal{T}'_ℓ by setting $\mathfrak{z}_e = \sum_{e \text{ lies on the unique } i-i' \text{ path in } \mathcal{T}_\ell} z_{ii',t_\ell}$ for all $e \in E(\mathcal{T}_\ell)$, and $\mathfrak{z}_e = \sum_{t \leq t_\ell} y_{i,t}$ for all $e = (i, v_i) \in E'$. Note that $\sum_{e \in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e) \mathfrak{z}_e = \sum_{e=(i,i')} d_{\mathcal{T}_\ell}(i, i') z_{e,t_\ell} = O(\log n) \sum_e d_e z_{e,t_\ell} = O(\log n) t_\ell$.
 - A1.3. Define $D_\ell = \{j : \tau_j \leq t_\ell\}$. For each client $j \in D_\ell$, we define the group $N'_j = \{v_i : i \in N_j\}$. We now compute a subtree T'_ℓ of \mathcal{T}'_ℓ as follows. We obtain $N := \log_2 m$ subtrees T''_1, \dots, T''_N . Each tree T''_r is obtained by executing the GKR algorithm $192 \log_2 n$ times on the tree \mathcal{T}'_ℓ with groups $\{N'_j\}_{j \in D_\ell}$, and taking the union of all the subtrees returned. Note that we may assume that $i \in T''_r$ iff $(i, v_i) \in T''_r$. Set T'_ℓ to be any tree in $\{T''_1, \dots, T''_N\}$ satisfying (i) $\sum_{(i,v_i) \in E(T'_\ell)} f_i \leq 40 \cdot 192 \log_2 n \sum_{(i,v_i) \in E'} f_i \mathfrak{z}_{i,v_i}$ and (ii) $\sum_{e \in E(\mathcal{T}'_\ell) \setminus E'} d_{\mathcal{T}_\ell}(e) \leq 40 \cdot 192 \log_2 n \sum_{e \in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e) \mathfrak{z}_e$; if no such tree exists, the algorithm fails.
 - A1.4. Now remove all the dummy edges from T'_ℓ , open all the facilities in the resulting tree, and convert the resulting tree into a tour Tour_ℓ traversing all the opened facilities. For every unconnected client j , we connect j to a facility in N_j if some such facility is open (and hence part of Tour_ℓ).
 - A2. Return the concatenation of the tours Tour_ℓ for $\ell = 0, 1, \dots, \mathcal{N}$ shortcutting whenever possible. This induces an ordering of the open facilities. If some client is left unconnected, we say that the algorithm has failed.
-

Analysis. The algorithm may fail in steps A1.3 and A2. Lemmas 2.4 and 2.5 bound the failure probability in each case by $1/\text{poly}(m)$. To bound the expected cost conditioned on success, it suffices to bound the expectation of the random variable that equals the cost incurred if the algorithm succeeds, and is 0 otherwise.

Since each client j is connected to a facility in N_j , the total connection cost is at most $4 \sum_j C_j^*$. To bound the remaining components of the cost, we first show that in any phase ℓ , the \mathfrak{z} -assignment defined above “covers” each group in $\{N'_j\}_{j \in D_\ell}$ to an extent of at least $\frac{2}{3}$ (Claim 2.3). Next, we show in Lemma 2.4 that for every client $j \in D_\ell$, the probability that a facility in N_j is included in the tree T'_ℓ , and hence opened in phase ℓ , is at least $\frac{5}{6} \cdot \frac{2}{3} = \frac{5}{9}$. The facility-cost incurred in a phase is $O(\log n) \sum_{i,t} f_i y_{i,t}$, and since $\tau_{\max} \leq T = \text{poly}(m)$, the number of phases is $O(\log m)$, so this bounds the facility-opening cost incurred. Also, since the probability that j is not connected (to a facility in N_j) in phase ℓ decreases geometrically (at a rate less than $1/2$) with ℓ when $t_\ell \geq \tau_j$, one can argue that (a) with very high probability (i.e., $1 - 1/\text{poly}(m)$), each client j is connected to some facility in N_j , and (b) the expected latency-cost of j is at most $O(\log n) \sum_{e \in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e) \mathfrak{z}_e = O(\log^2 n) \tau_j$.

Claim 2.3 Consider any phase ℓ . For any subset S of nodes of the corresponding tree \mathcal{T}'_ℓ with $r \notin S$, and any $N'_j \subseteq S$ where $j \in D_\ell$, we have $\mathfrak{z}(\delta(S)) \geq \sum_{i \in N_j, t \leq t_\ell} x_{ij,t} \geq 2/3$ (where $\delta(S)$ denotes $\delta_{\mathcal{T}'_\ell}(S)$).

Proof : Let $R = S \cap V(\mathcal{T}_\ell)$, and $Y \subseteq \mathcal{F}$ be the set of leaves in R . Then $\delta(S) = \delta_{\mathcal{T}_\ell}(R) \cup (\delta(S) \cap E')$. Let $\delta_E(Y) = \{(i, i') \in E : |\{i, i'\} \cap Y| = 1\}$. Observe that $\mathfrak{z}(\delta_{\mathcal{T}_\ell}(R)) \geq \sum_{e \in \delta_E(Y)} z_{e,t_\ell}$. This is simply because if we send z_{ii',t_ℓ} flow along the unique (i, i') path in \mathcal{T}_ℓ for every $(i, i') \in \delta_E(Y)$, then we obtain a flow between Y and $\mathcal{F} \cup \{r\} \setminus Y$ respecting the capacities $\{\mathfrak{z}_e\}_{e \in E(\mathcal{T}_\ell)}$, and of value equal to the RHS above. Thus, the inequality follows because the capacity of any cut containing Y must be at least the value of the flow. Since (x, y, z) satisfies (3), we further have that $\sum_{e \in \delta_E(Y)} z_{e,t_\ell} \geq \sum_{i \in Y, t \leq t_\ell} x_{ij,t}$. So

$$\mathfrak{z}(\delta(S)) \geq \sum_{e \in \delta_{\mathcal{T}_\ell}(R)} \mathfrak{z}_{e,t_\ell} + \sum_{(i,v_i) \in \delta(S): i \notin Y, t \leq t_\ell} (\sum y_{i,t}) \geq \sum_{i \in Y, t \leq t_\ell} x_{ij,t} + \sum_{i \in N_j \setminus Y, t \leq t_\ell} x_{ij,t} \geq \sum_{i \in N_j, t \leq t_\ell} x_{ij,t} \geq \frac{2}{3}.$$

Lemma 2.4 In any phase ℓ , with probability $1 - 1/\text{poly}(m)$, we obtain the desired tree T'_ℓ in step A1.3. Moreover, $\Pr[T'_\ell \cap N'_j \neq \emptyset] \geq 5/9$ for all $j \in D_\ell$.

Proof : Consider any tree T''_r obtained by executing the GKR algorithm $192 \log_2 n$ times and taking the union of the resulting subtrees. For brevity, we denote $\sum_{(i,v_i) \in E(T''_r)} f_i$ by $F(T''_r)$, and $\sum_{e \in E(T''_r) \setminus E'} d_{\mathcal{T}_\ell}(e)$ by $d_\ell(T''_r)$.

For $j \in D_\ell$, let \mathbf{E}_j^r denote the event that $T''_r \cap N'_j$ is non-empty. By Theorem 2.2, we have $\Pr[\mathbf{E}_j^r] \geq 1 - \exp(-\frac{\nu_j}{64 \log_2 n} \cdot 192 \log n) \geq 1 - e^{-3\nu_j} \geq (1 - e^{-3})\nu_j \geq 11/18$. We also have that $\mathbb{E}[F(T''_r)] \leq 192 \log_2 n \sum_{(i,v_i) \in E'} f_i \mathfrak{z}_{i,v_i}$ and $\mathbb{E}[d_\ell(T''_r)] \leq 192 \log_2 n \sum_{e \in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e) \mathfrak{z}_e$. Let \mathbf{F}^r and \mathbf{D}^r denote respectively the events that $F(T''_r) \leq 40 \cdot 192 \log_2 n \sum_{(i,v_i) \in E'} f_i \mathfrak{z}_{i,v_i}$, and $d_\ell(T''_r) \leq 40 \cdot 192 \log_2 n \sum_{e \in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e) \mathfrak{z}_e$. By Markov's inequality each event happens with probability at least $39/40$. Thus, for any $j \in D_\ell$, we get that

$$\Pr[\mathbf{E}_j^r | (\mathbf{F}^r \wedge \mathbf{D}^r)] \geq \Pr[\mathbf{E}_j^r \wedge \mathbf{F}^r \wedge \mathbf{D}^r] \geq 1 - (7/18 + 1/40 + 1/40) > 5/9. \quad (4)$$

Now the probability that $(\mathbf{F}^r \wedge \mathbf{D}^r)^c$ happens for all $r = 1, \dots, N$ is at most $(2/40)^N \leq 1/m^4$. Hence, with high probability, there is some tree $T'_\ell := T_r$ such that both \mathbf{F}^r and \mathbf{D}^r hold, and (4) shows that $\Pr[T'_\ell \cap N'_j \neq \emptyset] \geq 5/9$ for all $j \in D_\ell$. ■

Lemma 2.5 The probability that a client j is not connected by the algorithm is at most $1/m^4$. Let L_j be the random variable equal to j 's latency-cost if the algorithm succeeds and 0 otherwise. Then $\mathbb{E}[L_j] = O(\log^2 n) t_{\ell_j}$, where $\ell_j = \lceil \log_2 \tau_j \rceil$ is the smallest ℓ such that $t_\ell \geq \tau_j$.

Proof : Let P_j be the random variable denoting the phase in which j gets connected; let $P_j := \mathcal{N} + 1$ if j remains unconnected. We have $\Pr[P_j \geq \ell] \leq \left(\frac{4}{9}\right)^{(\ell - \ell_j)}$ for $\ell \geq \ell_j$. The algorithm proceeds for at least $4 \log_2 m$ phases after phase ℓ_j , so $\Pr[j \text{ is not connected after } \mathcal{N} \text{ phases}] \leq 1/m^4$. Now,

$$L_j \leq \sum_{\ell \leq P_j} d(\text{Tour}_\ell) \leq 2 \sum_{\ell \leq P_j} \sum_{e \in E(T'_\ell) \setminus E'} d_{T'_\ell}(e) = O(\log n) \sum_{\ell \leq P_j} \sum_{e \in E(T'_\ell)} d_{T'_\ell}(e) \mathbf{1}_e = O(\log^2 n) \sum_{\ell \leq P_j} t_\ell$$

$$\text{so } \mathbb{E}[L_j] = O(\log^2 n) \sum_{\ell=0}^{\mathcal{N}} \Pr[P_j \geq \ell] \cdot t_\ell \leq O(\log^2 n) \left[\sum_{\ell=0}^{\ell_j} t_\ell + \sum_{\ell > \ell_j} t_\ell \cdot \left(\frac{4}{9}\right)^{(\ell - \ell_j)} \right] = O(\log^2 n) t_{\ell_j}. \quad \blacksquare$$

Theorem 2.6 *Algorithm 1 succeeds with probability $1 - 1/\text{poly}(m)$, and returns a solution of expected cost $O(\log n \cdot \max\{\log n, \log m\}) \cdot OPT$.*

Proof : Lemmas 2.4 and 2.5 show that the failure probability is $1/\text{poly}(m)$. Let Y denote the cost incurred if the algorithm succeeds, and 0 otherwise. Since $t_{\ell_j} \leq 2\tau_j = O(L_j^*)$ for each j , we have $\mathbb{E}[Y] = O(\log n \cdot \max\{\log n, \log m\}) \cdot OPT$ by Lemma 2.5 and the preceding arguments. \blacksquare

Removing the assumption $\mathbf{T} = \text{poly}(m)$. We first argue that although (P) has a pseudopolynomial number of variables, one can compute a near-optimal solution to it in polynomial time (Lemma 2.7) by considering only (integer) time-values that are powers of $(1 + \epsilon)$ (roughly speaking). Given $\epsilon > 0$, define $\mathbf{T}_r = \lceil (1 + \epsilon)^r \rceil$, and let $\mathbf{TS} := \{\mathbf{T}_0, \mathbf{T}_1, \dots, \mathbf{T}_k\}$ where k is the smallest integer such that $\mathbf{T}_k \geq \min\{n, m\} d_{\max}$. Define $\mathbf{T}_{-1} = 0$. Let $(P)_{\mathbf{TS}}$ denote (P) when t ranges over \mathbf{TS} .

Lemma 2.7 *For any $\epsilon > 0$, we can obtain a solution to (P) of cost at most $(1 + \epsilon)OPT$ in time $\text{poly}(\text{input size}, 1/\epsilon)$.*

Proof : We prove that the optimal value of $(P)_{\mathbf{TS}}$ is at most $(1 + \epsilon)OPT$. Since the size of $(P)_{\mathbf{TS}}$ is $\text{poly}(\text{input size}, 1/\epsilon)$ this proves the lemma.

We transform (x, y, z) , an optimal solution to (P) to a feasible solution (x', y', z') to $(P)_{\mathbf{TS}}$ of cost at most $(1 + \epsilon)OPT$. (In fact, the facility-opening and connection-costs remain unchanged, and the latency-cost blows up by a $(1 + \epsilon)$ -factor.) z' is simply a restriction of z to the times in \mathbf{TS} , that is, $z'_{e,t} = z_{e,t}$ for each $e, t \in \mathbf{TS}$. Set $x'_{ij,1} = x_{ij,1}$, $y'_{i,1} = y_{i,1}$ for all i and j . For each $\ell = 1, \dots, k$, facility i , client j , we set $x'_{ij, \mathbf{T}_\ell} = \sum_{t=\mathbf{T}_{\ell-1}+1}^{\mathbf{T}_\ell} x_{ij,t}$ and $y'_{i, \mathbf{T}_\ell} = \sum_{t=\mathbf{T}_{\ell-1}+1}^{\mathbf{T}_\ell} y_{i,t}$. It is clear that $\sum_{t \in \mathbf{TS}} x'_{ij,t} = \sum_t x_{ij,t}$ and $\sum_{t \in \mathbf{TS}} y'_{i,t} = \sum_t y_{i,t}$ for all i and j , and moreover for any $\mathbf{T}_\ell \in \mathbf{TS}$, we have $\sum_{t \in \mathbf{TS}: t \leq \mathbf{T}_\ell} x'_{ij,t} = \sum_{t \leq \mathbf{T}_\ell} x_{ij,t}$. It follows that (x', y', z') is a feasible solution to $(P)_{\mathbf{TS}}$ and $\sum_{i,t \in \mathbf{TS}} f_i y'_{i,t} = \sum_{i,t} f_i y_{i,t}$, $\sum_{j,i,t \in \mathbf{TS}} c_{ij} x'_{ij,t} = \sum_{j,i,t} c_{ij} x_{ij,t}$. To bound the latency cost, note that for any $t > \mathbf{T}_{\ell-1}$, we have $\mathbf{T}_\ell \leq (1 + \epsilon)t$, so for any client j and facility i , $\sum_{t \in \mathbf{TS}} t x'_{ij,t} \leq x_{ij,1} + (1 + \epsilon) \sum_{t > 1} t x_{ij,t} \leq (1 + \epsilon) \sum_t t x_{ij,t}$. Thus, $\sum_{j,i,t \in \mathbf{TS}} t x'_{ij,t} \leq (1 + \epsilon) \sum_{j,i,t} t x_{ij,t}$. \blacksquare

Let (x', y', z') denote an optimal solution to $(P)_{\mathbf{TS}}$. The only changes to Algorithm 1 are in the definition of the time t_ℓ and the number of phases \mathcal{N} . (Of course we now work with (x', y', z') , and C_j^* , L_j^* are defined in terms of (x', y', z') now.) The idea is to define t_ℓ so that one can “reach” the τ_j of every client j in $O(\log m)$ phases; thus, one can terminate in $O(\log m)$ phases and thereby obtain the same approximation on the facility-opening cost. Let $\bar{L} = (\sum_j L_j^*)/m = (\sum_{j,i,t \in \mathbf{TS}} t x'_{ij,t})/m$. For $x \leq \mathbf{T}_k$, define $\mathbf{TS}(x)$ to be the earliest time in \mathbf{TS} that is at least x ; if $x \geq \mathbf{T}_k$, define $\mathbf{TS}(x) := \mathbf{T}_k$. Note that $\mathbf{TS}(x) \leq (1 + \epsilon)x$ for all $x \geq 0$. We now define $t_\ell = \mathbf{TS}(\bar{L} \cdot 2^\ell)$, and set the number of phases to $\mathcal{N} := \lceil \log_2(2\tau_{\max}/\bar{L}) + 4 \log_2 m \rceil$. Note that since $\tau_j = O(L_j^*)$, we have $\mathcal{N} = O(\log m)$.

Theorem 2.8 *For any $\epsilon > 0$, Algorithm 1 with the above modifications succeeds with high probability and returns a solution of expected cost $O(\log n \max\{\log n, \log m\})(1 + \epsilon)OPT$.*

Proof : The analysis of the facility-opening and connection-cost is exactly as before (since $\mathcal{N} = O(\log m)$). Define ℓ_j as the smallest ℓ such that $t_\ell \geq \tau_j$. Note that $\ell_j \leq \lfloor \log_2(2\tau_j/\bar{L}) \rfloor$ (this holds even when $\tau_j \leq \bar{L}$). Hence, the probability that j is not connected after \mathcal{N} phases is at most $1/m^4$. So as before, the failure probability is at most $1/\text{poly}(m)$. We have $\sum_{\ell \leq \ell_j} t_\ell \leq (1 + \epsilon)\bar{L} \sum_{\ell \leq \ell_j} 2^\ell \leq 2(1 + \epsilon)t_{\ell_j}$, and $\sum_{\ell > \ell_j} t_\ell \left(\frac{4}{9}\right)^{(\ell - \ell_j)} = O(t_{\ell_j})$. Thus, the inequalities involving L_j and $E[L_j]$ in Lemma 2.5 are still valid, and we obtain the same bound on $E[L_j]$ as in Lemma 2.5. Note that $t_{\ell_j} \leq 2(1 + \epsilon)\tau_j = O(L_j^*)$ when $\tau_j \geq \bar{L}$. Thus, $\sum_j E[L_j] \leq O(\log^2 n) [m \cdot \bar{L} + \sum_{j: \tau_j > t_0} O(L_j^*)] = O(\log^2 n)L^*$. ■

Inapproximability of MLUFL. We argue that any improvement in the guarantee obtained in Theorem 2.6 would yield an improvement in the approximation factor for GST. We reduce GST to MGL, the special case of MLUFL mentioned in Section 1, where we have groups $G_j \subseteq \mathcal{F}$ and the goal is to order the the facilities so as to minimize the sum of the covering times of the groups. (Note that Theorem 2.6 implies an $O(\log^2 n)$ -approximation for MGL.) Recall that we may assume that the groups in MGL are disjoint, in which case the connection costs form a metric.

Theorem 2.9 *Given a $\rho_{n,m}$ -approximation algorithm for MGL with (at most) n nodes and m groups, we can obtain an $O(\rho_{n,m} \log m)$ -approximation algorithm for GST with n nodes and m groups. Thus, the polylogarithmic inapproximability of GST [22] implies that MGL, and hence MLUFL even with metric connection costs, cannot be approximated to a factor better than $\Omega(\log m)$, even when the time-metric arises from a hierarchically well-separated tree, unless $\text{NP} \subseteq \text{ZTIME}(n^{\text{polylog}(n)})$.*

Our proof of the above theorem is LP-based and is deferred to Appendix A. Gupta et al. [21] independently arrived at the above theorem via a combinatorial proof.

2.2 MLUFL with related metrics

Here, we consider the MLUFL problem when the facilities, clients, and the root r are located in a common metric space that defines the connection-cost metric (on $\mathcal{F} \cup \mathcal{D} \cup \{r\}$), and we have $d_{uv} = c_{uv}/M$ for all $u, v \in \mathcal{F} \cup \mathcal{D} \cup \{r\}$. We call this problem, *related MLUFL*, and design an $O(1)$ -approximation algorithm for it.

The algorithm follows a similar outline as Algorithm 1. As before, we build the tour on the open facilities by concatenating tours obtained by “Eulerifying” trees rooted at r of geometrically increasing length. At a high level, the improvement in the approximation arises because one can now obtain these trees without resorting to Theorems 2.1 and 2.2 and losing $O(\log n)$ -factors in process. Instead, since the d - and c -metrics are related, we can obtain a group Steiner tree on the relevant groups by using a Steiner tree algorithm (in a manner similar to the LP-rounding algorithms in [27, 20]). We now define $N_j = \{i : \sum_t x_{ij,t} > 0, c_{ij} \leq 3C_j^*\}$, and $\tau_j = 6L_j^*$. Ideally, in each phase ℓ , we want to connect the N_j groups for all j such that $\tau_j \leq t_\ell := 2^\ell$. But to obtain a low-cost solution, we do a facility-location-style clustering of the set of clients with $\tau_j \leq t_\ell$ and build a tree \mathcal{T}_ℓ that connects the N_j s of only the cluster centers: we contract these N_j s and build an MST (in the d -metric) on them, and then connect each N_j internally (in the d -metric) using intracluster edges incident on j . *Here we crucially exploit the fact that the d - and c -metrics are related.*

Deciding which facilities to open is tricky because groups N_j and N_k created in different phases could overlap, and we could have $C_j^* \ll C_k^*$ but $\tau_j \gg \tau_k$; so if we open $i \in N_j$ and use this to also serve k , then we must connect i to some \mathcal{T}_ℓ (without increasing its d -cost by much) where $t_\ell = O(\tau_k)$. We consider the collection \mathcal{C} of cluster centers created in all the phases, and pick a maximal subset $\mathcal{C}' \subseteq \mathcal{C}$ that yields disjoint clusters by greedily considering clusters in increasing C_j^* order. We open the cheapest facility in N_j for all $j \in \mathcal{C}'$, and attach it to the tree \mathcal{T}_ℓ , where ℓ is the *earliest* phase such that there is some cluster N_k created

in that phase that was removed (from \mathcal{C}) when N_j was included in \mathcal{C}' (because $N_k \cap N_j \neq \emptyset$). Since d and c are related, one can bound the resulting increase in the d -cost of \mathcal{T}_ℓ . Finally, we convert these augmented \mathcal{T}_ℓ -trees to tours and concatenate these tours.

We now describe the algorithm in detail. We have not sought to optimize the approximation ratio. When we refer to an edge or a node below, we mean an edge or node of the complete graph on $\mathcal{F} \cup \mathcal{D} \cup \{r\}$. Recall that $N_j = \{i : \sum_t x_{ij,t} > 0, c_{ij} \leq 3C_j^*\}$, and $\tau_j = 6L_j^*$. So $\sum_{i \in N_j} \sum_t x_{ij,t} \geq \frac{2}{3}$, and $\sum_{i \in N_j, t \leq \tau_j} x_{ij,t} \geq \frac{1}{2}$.

- R1. For each time $t_\ell = 2^\ell$, where $\ell = 0, 1, \dots, \lceil \log T \rceil$, we do the following. Define $D_\ell = \{j : \tau_j \leq t_\ell\} \setminus (\bigcup_{0 \leq \ell' < \ell} \mathcal{C}_{\ell'})$ (where the union of an empty collection is \emptyset).
 - **(Clustering)** We cluster the facilities in $\bigcup_{j \in D_\ell} N_j$ as follows. We pick $j \in D_\ell$ with smallest C_j^* value and form a cluster around j consisting of the facilities in N_j . For every client $k \in D_\ell$ (including j) such that $c_{jk} \leq 30C_k^*$ (note that $C_k^* \geq C_j^*$), we remove k from D_ℓ , set $\sigma(k) = j$, and recurse on the remaining clients in D_ℓ until no client is left in D_ℓ . Let \mathcal{C}_ℓ denote the set of cluster centers (i.e., $\{j \in D_\ell : \sigma(j) = j\}$). Note that for two clients j and j' in \mathcal{C}_ℓ , $N_j \cap N_{j'}$ is \emptyset .
 - **(Building a group Steiner tree \mathcal{T}_ℓ on $\{N_j\}_{j \in \mathcal{C}_\ell}$)** We contract the clusters N_j for $j \in \mathcal{C}_\ell$ into supernodes, and build a minimum spanning tree (MST) \mathcal{T}_ℓ'' connecting r and these supernodes. Next, we uncontract the supernodes, and for each $j \in \mathcal{C}_\ell$, we add edges joining j to every facility $i \in N_j$ that has an edge incident to it in \mathcal{T}_ℓ'' . This yields the tree \mathcal{T}_ℓ .
- R2. **(Opening facilities)** Let $\mathcal{C} = \bigcup_\ell \mathcal{C}_\ell$. (Note that a client appears in at most one of the \mathcal{C}_ℓ sets.) We cannot open a facility in every cluster centered around a client in \mathcal{C} , since for $j \in \mathcal{C}_\ell$ and $k \in \mathcal{C}_{\ell'}$, N_j and N_k need not be disjoint. So we select a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that for any two j, k in \mathcal{C}' , the sets N_j and N_k are disjoint. This is done as follows. We initialize $\mathcal{C}' \leftarrow \emptyset$. Pick the client $j \in \mathcal{C}$ with smallest C_j^* value and add it to \mathcal{C}' . We delete from \mathcal{C} every client $k \in \mathcal{C}$ (including j) such that $N_k \cap N_j \neq \emptyset$, setting $\text{nbr}(k) = j$, and recurse on the remaining set of clients until no client is left in \mathcal{C} .
Consider each $j \in \mathcal{C}'$. We open the facility $i \in N_j$ with smallest f_i . Let ℓ be the smallest index such that there is some $k \in \mathcal{C}_\ell$ with $\text{nbr}(k) = j$. We connect i to \mathcal{T}_ℓ by adding the *facility edge* (i, k) to it. Let \mathcal{T}_ℓ' denote \mathcal{T}_ℓ augmented by all such facility edges.
- R3. We obtain a tour connecting all the open facilities, by converting each tree \mathcal{T}_ℓ' into a tour, and concatenating the tours for $\ell = 0, \dots, \lceil T \rceil$ (in that order).
- R4. For every client $j \in \mathcal{C}$, we assign j to the facility opened from $N_{\text{nbr}(j)}$. For every client $j \notin \mathcal{C}$, we assign j to the same facility as $\sigma(j)$.

Theorem 2.10 *For related MLUFL, one can round (x, y, z) to get a solution with facility-opening cost at most $\frac{3}{2} \sum_{i,t} f_i y_{i,t}$, where each client j incurs connection-cost at most $39C_j^*$ and latency-cost at most $64\tau_j = 384L_j^*$. Thus, we obtain an $O(1)$ -approximation algorithm for related MLUFL.*

Proof : The clusters N_j for clients $j \in \mathcal{C}'$ are disjoint; each such cluster has facility weight $\sum_{i \in N_j} \sum_t y_{i,t} \geq \frac{2}{3}$, and we open the cheapest facility in the cluster.

Consider a client j , and let it be assigned to facility i . If $j \in \mathcal{C}'$, then $i \in N_j$, and we have $c_{ij} \leq 3C_j^*$. If $j \in \mathcal{C} \setminus \mathcal{C}'$ with $\text{nbr}(j) = k$, then $i \in N_k$ and there is some facility $i' \in N_j \cap N_k$. So $c_{ij} \leq c_{ik} + c_{i'k} + c_{i'j} \leq 2 \cdot 3C_k^* + 3C_j^* \leq 9C_j^*$. Finally, if $j \notin \mathcal{C}$ and $j' = \sigma(j)$, then $\sigma(j')$ is also assigned to i , so $c_{ij} \leq c_{ij'} + c_{jj'} \leq 9C_{j'}^* + 30C_j^* \leq 39C_j^*$.

We next bound $d(\mathcal{T}_\ell)$ and $d(\mathcal{T}_\ell')$ for any phase ℓ . For any client $j \in D_\ell$ and any node-set $S \supseteq N_j$, $r \notin S$, we have $\sum_{e \in \delta(S)} z_{e,t_\ell} \geq \sum_{i \in S, t \leq t_\ell} x_{ij,t} \geq \sum_{i \in N_j, t \leq \tau_j} x_{ij,t} \geq \frac{1}{2}$. Therefore, $(2z_{e,t_\ell})$ forms a fractional

Steiner tree of cost at most $2t_\ell$ on the supernodes and r , and hence, $d(\mathcal{T}_\ell'') \leq 4t_\ell$ since it is well known that the cost of an MST is at most twice the cost of a fractional solution to the Steiner-tree LP. For $j \in \mathcal{C}_\ell$, let \deg_j denote the degree of the cluster N_j in \mathcal{T}_ℓ'' . Observe that if e is an edge of \mathcal{T}_ℓ'' joining N_j and N_k (so $j, k \in \mathcal{C}_\ell$), then $c_e \geq 24 \max\{C_j^*, C_k^*\}$, since $30 \max\{C_j^*, C_k^*\} \leq c_{jk} \leq 3C_j^* + c_e + 3C_k^*$. So the d -cost of adding the additional edges to \mathcal{T}_ℓ'' is at most $\frac{1}{M} \cdot \sum_{j \in \mathcal{C}_\ell} \deg_j \cdot 3C_j^* \leq d(\mathcal{T}_\ell'')/4$, and hence, $d(\mathcal{T}_\ell) \leq 5t_\ell$.

Now consider the cost of adding facility edges to \mathcal{T}_ℓ in step R2. For each facility edge (i, k) added, we know that $i \in N_{\text{nbr}}(k)$, $k \in \mathcal{C}_\ell$, and k is assigned to i . So we have $c_{ik} \leq 9C_k^*$. Observe that each client $k \in \mathcal{C}_\ell$ is responsible for at most one such facility edge. So the d -cost of these facility edges is at most $\frac{1}{M} \cdot \sum_{j \in \mathcal{C}_\ell} 9C_j^* \leq \frac{1}{M} \sum_{j \in \mathcal{C}_\ell} \deg_j \cdot 9C_j^* \leq \frac{3}{4} \cdot d(\mathcal{T}_\ell'')$. Thus, $d(\mathcal{T}_\ell') \leq d(\mathcal{T}_\ell) + \frac{3}{4} \cdot d(\mathcal{T}_\ell'') \leq 8t_\ell$.

Finally, we prove that the latency cost of any client j is at most $64\tau_j = 384L_j^*$. Let j be assigned to facility i . Let ℓ be the smallest index such that $j \in D_\ell$, so $t_\ell \leq 2\tau_j$. We first argue that if i is part of the tree \mathcal{T}_ℓ' , then $\ell' \leq \ell$. If $j \in \mathcal{C}$, this follows since we know that $i \in N_{\text{nbr}}(j)$ and $\ell' = \min\{r : \exists k \in \mathcal{C}_r \text{ with } \text{nbr}(k) = \text{nbr}(j)\}$. If $j \notin \mathcal{C}$, then we know that $\sigma(j) \in \mathcal{C}_\ell$ is also assigned to i , and so by the preceding argument, we again have that $\ell' \leq \ell$. Thus, the latency-cost of j is bounded by $\sum_{r=0}^{\ell'} 2d(\mathcal{T}_r') \leq 16 \sum_{r=0}^{\ell'} t_r \leq 32t_{\ell'} \leq 64\tau_j$. \blacksquare

2.3 MLUFL with a uniform time-metric

We now consider the special case of MLUFL, referred to as *uniform MLUFL*, where the time-metric d is uniform, that is, $d_{ii'} = 1$ for all $i, i' \in \mathcal{F} \cup \{r\}$. When the connection costs form a metric, we call it the *metric uniform MLUFL*. We consider the following simpler LP-relaxation of the problem, where the time t now ranges from 1 to n .

$$\begin{aligned} \min \quad & \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} (c_{ij} + t) x_{ij,t} \quad \text{subject to} \quad (\text{Unif-P}) \\ & \sum_{i,t} x_{ij,t} \geq 1 \quad \forall j; \quad x_{ij,t} \leq y_{i,t} \quad \forall i, j, t; \quad \sum_i y_{i,t} \leq 1 \quad \forall t; \quad x_{ij,t}, y_{i,t} \geq 0 \quad \forall i, j, t. \end{aligned}$$

Let (x, y) be an optimal solution to (Unif-P), and OPT be its value. Let $C_j^* = \sum_{i,t} c_{ij} x_{ij,t}$, $L_j^* = \sum_{i,t} t x_{ij,t}$. As stated in the introduction, uniform MLUFL generalizes: (i) set cover, when the facility and connection costs are arbitrary; (ii) MSSC, when the facility costs are zero (ZFC MLUFL); and (iii) metric UFL, when the connection costs form a metric. We obtain approximation bounds for uniform MLUFL, ZFC MLUFL, and metric MLUFL (Theorems 2.11 and 2.14) that complement these observations.

The main result of this section is Theorem 2.12, which shows that a ρ_{UFL} -approximation algorithm for UFL and a γ -approximation algorithm for ZFC MLUFL (with metric connection costs) can be combined to yield a $(\rho_{\text{UFL}} + 2\gamma)$ -approximation algorithm for metric uniform MLUFL. Taking $\rho_{\text{UFL}} = 1.5$ [8] and $\gamma = 9$ (part (ii) of Theorem 2.11), we obtain a 19.5-approximation algorithm. We improve this to 10.773 by using a more refined version of Theorem 2.12, which capitalizes on the asymmetric approximation bounds that one can obtain for different portions of the total cost in UFL and ZFC MLUFL.

We note that by considering each (i, t) as a facility, since the connection costs $c_{ij} + t$ form a metric, one can view metric MLUFL as a variant of metric UFL, and use the ideas in [3] to devise an $O(1)$ -approximation for this variant. We instead present our alternate algorithm based on the reduction in Theorem 2.12, because this reduction is quite robust and versatile. In particular, it allows us to: (a) handle certain extensions of the problem, e.g., the setting where we have non-uniform latency costs (see Section 4), for which the above reduction fails since we do not necessarily obtain metric connection costs, and (b) devise algorithms for the uniform latency versions of other facility location problems, where the cost of a facility does not depend on the client-set assigned to it (so one can assign a client to any open facility freely without affecting the facility costs). For instance, consider uniform MLUFL with the restriction that at most k facilities may be

opened: our technique yields a $(\rho_{k\text{Med}} + 2\gamma)$ approximation for this problem, using a $\rho_{k\text{Med}}$ -approximation for k -median.

Theorem 2.11 *One can obtain:*

- (i) an $O(\ln m)$ -approximation algorithm for uniform MLUFL with arbitrary facility- and connection- costs.
- (ii) a solution of cost at most $\frac{1}{1-\alpha} \sum_j C_j^* + \frac{4}{\alpha} \lceil \frac{1}{\alpha} \rceil \sum_j L_j^*$ for ZFCMLUFL, for any parameter $\alpha \in (0, 1)$. Thus, setting $\alpha = \frac{8}{9}$, yields solution of cost at most $9 \cdot \text{OPT}$.

We defer the proof of Theorem 2.11 to the end of the section, and focus first on detailing the aforementioned reduction.

Theorem 2.12 *Given a ρ_{UFL} -approximation algorithm \mathcal{A}_1 for UFL, and a γ -approximation algorithm \mathcal{A}_2 for uniform ZFC MLUFL, one can obtain a $(\rho_{\text{UFL}} + 2\gamma)$ -approximation algorithm for metric uniform MLUFL.*

Proof : Let \mathcal{I} denote the metric uniform MLUFL instance, and O^* denote the cost of an optimal integer solution. Let \mathcal{I}_{UFL} be the UFL instance obtained from \mathcal{I} by ignoring the latency costs, and \mathcal{I}_{ZFC} be the ZFC MLUFL instance obtained from \mathcal{I} by setting all facility costs to zero. Let O_{UFL}^* and O_{ZFC}^* denote respectively the cost of the optimal (integer) solutions to these two instances. Clearly, we have $O_{\text{UFL}}^*, O_{\text{ZFC}}^* \leq O^*$. We use \mathcal{A}_1 to obtain a near-optimal solution to \mathcal{I}_{UFL} : let F_1 be the set of facilities opened and let $\sigma_1(j)$ denote the facility in F_1 to which client j is assigned. So we have $\sum_{i \in F_1} f_i + \sum_j c_{\sigma_1(j)j} \leq \rho_{\text{UFL}} \cdot O_{\text{UFL}}^*$. We use \mathcal{A}_2 to obtain a near-optimal solution to \mathcal{I}_{ZFC} : let F_2 be the set of open facilities, $\sigma_2(j)$ be the facility to which client j is assigned, and $\pi(i)$ be the position of facility i . So we have $\sum_j (c_{\sigma_2(j)j} + \pi(\sigma_2(j))) \leq \gamma \cdot O_{\text{ZFC}}^*$.

We now combine these solutions as follows. For each facility $i \in F_2$, let $\mu(i) \in F_1$ denote the facility in F_1 that is nearest to i . We open the set $F = \{\mu(i) : i \in F_2\}$ of facilities. The position of facility $i \in F$ is set to $\min_{i' \in F_2: \pi(i')=i} \pi(i')$. Each facility in F is assigned a distinct position this way, but some positions may be vacant. Clearly we can always convert the above into a proper ordering of F where each facility $i \in F$ occurs at position $\kappa(i) \leq \min_{i' \in F_2: \pi(i')=i} \pi(i')$. Finally, we assign each client j to the facility $\phi(j) = \mu(\sigma_2(j)) \in F$. Note that $\kappa(\phi(j)) \leq \pi(\sigma_2(j))$ (by definition). For a client j , we now have $c_{\phi(j)j} \leq c_{\sigma_2(j)\mu(\sigma_2(j))} + c_{\sigma_2(j)j} \leq c_{\sigma_2(j)\sigma_1(j)} + c_{\sigma_2(j)j} \leq c_{\sigma_1(j)j} + 2c_{\sigma_2(j)j}$. Thus, the total cost of the resulting solution is at most $\sum_{i \in F_1} f_i + \sum_j (c_{\sigma_1(j)j} + 2c_{\sigma_2(j)j} + \pi(\sigma_2(j))) \leq (\rho_{\text{UFL}} + 2\gamma) \cdot O^*$. ■

We call an algorithm a (ρ_f, ρ_c) -approximation algorithm for UFL if given an LP-solution to UFL with facility- and connection- costs F^* and C^* respectively, it returns a solution of cost at most $\rho_f F^* + \rho_c C^*$. Similarly, we say that an algorithm is a (γ_c, γ_l) -approximation algorithm for uniform ZFC MLUFL if given a solution to (P) with connection- and latency- costs C^* and L^* respectively, it returns a solution of cost at most $\gamma_c C^* + \gamma_l L^*$. The proof of Theorem 2.12 easily yields the following more general result.

Corollary 2.13 *One can combine a (ρ_f, ρ_c) -approximation algorithm for UFL, and a (γ_c, γ_l) -approximation algorithm for uniform ZFC MLUFL, to obtain a solution of cost at most $\max\{\rho_f, \rho_c + 2\gamma_c, \gamma_l\} \cdot \text{OPT}$.*

Proof : The proof mimics the proof of Theorem 2.12. The only new observation is that (x, y) yields an LP-solution to (i) \mathcal{I}_{UFL} with facility cost $\sum_{i,t} f_i y_{i,t}$ and connection cost $\sum_{j,i,t} c_{ij} x_{ij,t}$; and (ii) \mathcal{I}_{ZFC} with connection cost $\sum_{j,i,t} c_{ij} x_{ij,t}$ and latency cost $\sum_{j,i,t} t x_{ij,t}$. Thus, applying the construction in the proof of Theorem 2.12 yields a solution of total cost at most

$$\rho_f \sum_{i,t} f_i y_{i,t} + (\rho_c + 2\gamma_c) \sum_{j,i,t} c_{ij} x_{ij,t} + \gamma_l \sum_{j,i,t} t x_{ij,t} \leq \max\{\rho_f, \rho_c + 2\gamma_c, \gamma_l\} \cdot \text{OPT}. \quad \blacksquare$$

Combining the $(\frac{\ln(1/\beta)}{1-\beta}, \frac{3}{1-\beta})$ -approximation algorithm for UFL [29] with the $(\frac{1}{1-\alpha}, \frac{4}{\alpha} \lceil \frac{1}{\alpha} \rceil)$ -approximation algorithm for ZFC MLUFL (part (ii) of Theorem 2.11) gives the following result.

Theorem 2.14 For any $\alpha, \beta \in (0, 1)$, one can obtain a solution of cost $\max\{\frac{\ln(1/\beta)}{1-\beta}, \frac{3}{1-\beta} + \frac{2}{1-\alpha}, \frac{4}{\alpha} \lceil \frac{1}{\alpha} \rceil\} \cdot \text{OPT}$. Thus, taking $\alpha = 0.7426$, $\beta = 0.000021$, we obtain a 10.773-approximation algorithm.

2.3.1 Proof of Theorem 2.11

The following lemma will often come in handy.

Lemma 2.15 Let (\hat{x}, \hat{y}) be a solution satisfying $\sum_i \hat{y}_{i,t} \leq k$ for every time t , where $k \geq 0$ is an integer, and all the other constraints of (Unif-P). Then, one can obtain a feasible solution (x', y') to (Unif-P) such that (i) $\sum_{i,t} f_i y'_{i,t} = \sum_{i,t} f_i \hat{y}_{i,t}$; (ii) $\sum_{j,i,t} c_{ij} x'_{ij,t} = \sum_{j,i,t} c_{ij} \hat{x}_{ij,t}$; and (iii) $\sum_{j,i,t} t x'_{ij,t} \leq k \cdot \sum_{j,i,t} t \hat{x}_{ij,t}$.

Proof : For each time t , define $S_t = \{(i, t) : \hat{y}_{i,t} > 0\}$. The idea is to simply “spread out” the S_t sets. Let $S_0 = \bigcup_t S_t$ be an ordered list where all the (\cdot, t) pairs are listed before any $(\cdot, t+1)$ pair, and the pairs for a given t (i.e., $(i, t) \in S_t$) are listed in arbitrary order. Let $T_0 = \left\lceil \sum_{i,t} \hat{y}_{i,t} \right\rceil$. We divide the pairs in S_0 into T_0 groups as follows. Initialize $\ell \leftarrow 1$, $S \leftarrow S_0$. For a set A of (i, t) pairs, we define the \hat{y} -weight of A as $\hat{y}(A) = \sum_{(i,t) \in A} \hat{y}_{i,t}$, and the \hat{x}_j -weight of A as $\sum_{(i,t) \in A} \hat{x}_{ij,t}$. If $0 < \hat{y}(S) \leq 1$, then we set $G_\ell = S$ to end the grouping process. Otherwise, group G_ℓ includes all pairs of S , taken in order starting from the first pair, stopping when the total \hat{y} -weight of the included pairs becomes at least 1; all the included pairs are also deleted from S . If the \hat{y} -weight of G_ℓ now exceeds 1, then we split the last pair (i, t) into two copies: we include the first copy in G_ℓ and retain the second copy in S , and distribute $\hat{y}_{i,t}$ across the \hat{y} -weight of the two copies so that $\hat{y}(G_\ell)$ is now *exactly* 1. (Thus, the new \hat{y} -weight of S is precisely its old \hat{y} -weight -1 .) Also, for each client j , we distribute $\hat{x}_{ij,t}$ across the \hat{x}_j -weight of the two copies arbitrarily while maintaining that the \hat{x}_j -weight of each copy is at most its \hat{y} -weight. We update $\ell \leftarrow \ell + 1$, and continue in this fashion with the current (i.e., ungrouped) list of pairs S . Note that an (i, t) pair in S_0 may be split into at most two copies above (that lie in consecutive groups). To avoid notational clutter, we call both these copies (i, t) and use $\hat{y}_{i,t}^\ell$ to denote the \hat{y} -weight of the copy in group G_ℓ (which is equal to the original $\hat{y}_{i,t}$ if (i, t) is not split). Analogously, we use $\hat{x}_{ij,t}^\ell$ to denote the \hat{x}_j -weight of the copy of (i, t) in group G_ℓ .

For every facility i , client j , and $\ell = 1, \dots, T_0$, we set $y'_{i,\ell} = \sum_{t:(i,t) \in G_\ell} \hat{y}_{i,t}^\ell$ and $x'_{ij,\ell} = \sum_{t:(i,t) \in G_\ell} \hat{x}_{ij,t}^\ell$, so we have $x'_{ij,\ell} \leq y'_{i,\ell}$. It is clear that $\sum_\ell y'_{i,\ell} = \sum_t \hat{y}_{i,t}$ and $\sum_\ell x'_{ij,\ell} = \sum_t \hat{x}_{ij,t}$ for every facility i and client j . Thus, (x', y') is feasible to (Unif-P), and parts (i) and (ii) of the lemma hold. To prove part (iii), note that if (some copy of) (i, t) is in G_ℓ , then $\ell \leq kt$, since then we have $(\bigcup_{r=1}^{\ell-1} G_r) \subset \bigcup_{t'=1}^t S_{t'}$ and so $\ell - 1 < kt$. Thus, for any client j , we have

$$\sum_{i,\ell} \ell x'_{ij,\ell} = \sum_{i,\ell} \ell \left(\sum_{t:(i,t) \in G_\ell} \hat{x}_{ij,t}^\ell \right) = \sum_{i,t} \sum_{\ell:(i,t) \in G_\ell} \ell \hat{x}_{ij,t}^\ell \leq \sum_{i,t} kt \sum_{\ell:(i,t) \in G_\ell} \hat{x}_{ij,t}^\ell = k \cdot \sum_{i,t} t \hat{x}_{ij,t}.$$

Proof of part (i) of Theorem 2.11 : We round the LP-optimal solution (x, y) by using filtering followed by standard randomized rounding. Clearly, we may assume that $\sum_{i',t} x_{i',j,t} = 1$ and $y_{i,t} = \max_j x_{ij,t}$ for every i, j, t . Also, we may assume that if $\sum_i y_{i,t+1} > 0$, then $\sum_i y_{i,t} = 1$, because otherwise for some facility i and some $\epsilon > 0$, we may decrease $y_{i,t+1}$ by ϵ and increase $y_{i,t}$ by ϵ , and modify the $\{x_{ij,t+1}, x_{ij,t}\}_j$ values appropriately so as to maintain feasibility, without increasing the total cost. Define $N_j = \{(i, t) : c_{ij} + t \leq 2(C_j^* + L_j^*)\}$ for a client j . The algorithm is as follows.

U1. For each (i, t) , we set $Y_{i,t} = 1$ independently with probability $\min\{4 \ln m \cdot y_{i,t}, 1\}$.

U2. Considering each client j , if $\{(i, t) \in N_j : Y_{i,t} = 1\} = \emptyset$, then set $Y_{i_j,1} = 1$ where i_j is such that $f_{i_j} = \min_{(i,t) \in N_j} f_i$ (note that $(i_j, 1) \in N_j$). Let $S_j = \{(i, t) \in N_j : Y_{i,t} = 1\}$ (which is non-empty). Assign each client j to the (i, t) pair in S_j with minimum $c_{ij} + t$ value, i.e., set $X_{ij,t} = 1$.

U3. Let $K = \max_t \sum_i Y_{i,t}$. Use Lemma 2.15 to convert (X, Y) into a feasible integer solution to (Unif-P).

Let C_j and L_j denote respectively the connection cost and latency cost of client j in (X, Y) . We argue that (i) $\mathbb{E}[\sum_{i,t} f_i Y_{i,t}] = O(\ln m) \cdot \sum_{i,t} f_i y_{i,t}$, (ii) with probability 1, $C_j + L_j \leq 2(C_j^* + L_j^*)$ for every client j , and (iii) $K = O(\ln m)$ with high probability, and in expectation. The theorem then follows from Lemma 2.15.

For any client j , we have $\sum_{(i,t) \in N_j} x_{ij,t} \geq \frac{1}{2}$ (by Markov's inequality). Thus, $f_{ij} \leq 2 \sum_{(i,t) \in N_j} f_i y_{i,t}$ and $\Pr[\sum_{(i,t) \in N_j} Y_{i,t} = 0]$ is at most $e^{-4 \ln m \cdot \frac{1}{2}} = 1/m^2$. The expected cost of opening facilities in step U1 is clearly at most $4 \ln m \cdot \sum_{i,t} f_i y_{i,t}$. The expected facility-opening cost in step U2 is at most $\Pr[\text{facility is opened in step U2}] \cdot \sum_j f_{ij} \leq m \cdot \frac{1}{m^2} \cdot 2m \sum_{i,t} f_i y_{i,t}$. So $\mathbb{E}[\sum_{i,t} f_i Y_{i,t}] = O(\ln m) \cdot \sum_{i,t} f_i y_{i,t}$. Since we always open some (i, t) pair in N_j , we have $C_j + L_j \leq 2(C_j^* + L_j^*)$ for every client j .

Let $S = \{t : \sum_i y_{i,t} > 0\}$. Note that $|S| \leq 1 + \sum_{i,t} y_{i,t} \leq 1 + \sum_{j,i,t} x_{ij,t} = m + 1$. After step U1, we have $\mathbb{E}[\sum_i Y_{i,t}] \leq 4 \ln m$ for every time $t \in S$. Since the $Y_{i,t}$ random variables are independent, we also have $\Pr[\sum_i Y_{i,t} > 8 \ln m] \leq 1/m^2$ for all $t \in S$ (and also, $\mathbb{E}[\max_{t \in S} \sum_i Y_{i,t}] = O(\ln m)$). Thus, after step U2, we have $\Pr[\sum_i Y_{i,t} > 8 \ln m] \leq 1/m^2 + 1/m$ and $\Pr[\sum_i Y_{i,t} > 8 \ln m] \leq 1/m^2$ for all $t \in S$, $t > 1$. Hence, $\Pr[K > 8 \ln m] \leq 2/m$. (This also shows that $\mathbb{E}[K] = O(\ln m)$.) ■

Proof of part (ii) of Theorem 2.11 : We round (x, y) by applying filtering [23] followed by Lemma 2.15 to reduce the problem to a MSSC problem, and then use the result of Feige et al. [16] to obtain a near-optimal solution to this MSSC problem. Let

$$\min \sum_{j,t} t x_{j,t} \quad \text{s.t.} \quad \sum_t x_{j,t} \geq 1 \quad \forall j, \quad x_{j,t} \leq \sum_{S:j \in S} y_{S,t} \quad \forall j,t, \quad \sum_S y_{S,t} \leq 1 \quad \forall t, \quad x, y \geq 0. \quad (\text{P1})$$

denote the standard LP-relaxation of MSSC [16] (here j indexes the elements, S indexes the sets, and t indexes time). Feige et al. showed that given a solution (\hat{x}, \hat{y}) to (P1), one can obtain in polytime an integer solution of cost at most $4 \cdot \sum_{j,t} t \hat{x}_{j,t}$.

The rounding algorithm for ZFC MLUFL is as follows. Define $N_j = \{i : c_{ij} \leq C_j^*/(1 - \alpha)\}$, so $\sum_{i \in N_j, t} x_{ij,t} \geq \alpha$. For all i, j, t , set $\hat{y}_{i,t} = y_{i,t}/\alpha$, and $\hat{x}_{ij,t} = x_{ij,t}/\alpha$ if $i \in N_j$ and $\hat{x}_{ij,t} = 0$ otherwise. It is easy to see that (\hat{x}, \hat{y}) satisfies $\sum_i \hat{y}_{i,t} \leq \frac{1}{\alpha}$ for all t , and all the other constraints of (Unif-P). We use Lemma 2.15 to convert (\hat{x}, \hat{y}) to a feasible solution (x', y') to (Unif-P). Next, we extract a solution to (P1) from (x', y') . We identify facility i with the set $\{j : i \in N_j\}$, and set $\bar{x}_{j,t} = \sum_{i \in N_j} x'_{ij,t}$. Now (\bar{x}, y') is a feasible solution to (P1). Finally, we round (\bar{x}, y') to an integer solution. This yields the ordering $\tilde{y} = (\tilde{y}_{i,t})$ of the facilities. For each client j , if j is first covered by set i (so $i \in N_j$) at time (or position) t in the MSSC solution, then we set $\tilde{x}_{ij,t} = 1$.

Analysis. Since a client j is always assigned to a facility in N_j , the connection cost of j is bounded by $C_j^*/(1 - \alpha)$. To bound the latency cost, first we bound the cost of (\hat{x}, \hat{y}) . Since we modify the assignment of a client j by transferring weight from farther facilities to nearer ones, it is clear that $\sum_{i,t} c_{ij} \hat{x}_{ij,t} \leq \sum_{i,t} c_{ij} x_{ij,t}$. Also, clearly $\sum_{j,i,t} t \hat{x}_{ij,t} \leq \frac{1}{\alpha} \cdot \sum_{j,i,t} t x_{ij,t}$ and $\sum_i \hat{y}_{i,t} \leq \frac{1}{\alpha}$. Thus, applying Lemma 2.15 yields (x', y') satisfying $\sum_{j,i,t} t x'_{ij,t} \leq \lceil \frac{1}{\alpha} \rceil \cdot \frac{1}{\alpha} \cdot \sum_{j,i,t} t x_{ij,t}$. The result of [16] now implies that $\sum_{j,i,t} t \tilde{x}_{ij,t} \leq 4 \sum_{j,t} t \bar{x}_{j,t} = 4 \sum_{j,i,t} t x'_{ij,t} \leq \frac{4}{\alpha} \lceil \frac{1}{\alpha} \rceil \sum_{j,i,t} t x_{ij,t}$. ■

3 LP-relaxations and algorithms for the minimum-latency problem

In this section, we consider the minimum-latency (ML) problem and apply our techniques to obtain LP-based insights and algorithms for this problem. We give two LP-relaxations for ML with constant integrality

gap. The first LP (LP1) is a specialization of (P) to ML, and to bound its integrality gap, we only need the fact that the natural subtour elimination LP for TSP has constant integrality gap. The second LP (LP2_P) has exponentially-many variables, one for every path (or tree) of a given length bound, and the separation oracle for the dual problem corresponds to an (path- or tree-) orienteering problem. We prove that even a bicriteria approximation for the orienteering problem yields an approximation for ML while losing a constant factor. (The same relationship also holds between MGL and “group orienteering”.) As mentioned in the Introduction, we believe that our results shed new light on ML and opens up ML to new venues of attack. Moreover, as shown in Section 4 these LP-based techniques can easily be used to handle more general variants of ML, e.g., k -route ML with \mathcal{L}_p -norm latency-costs (for which we give the first approximation algorithm). We believe that our LP-relaxations are in fact (much) better than what we have accounted for, and conjecture that the integrality gap of both (LP1) and (LP2_P) is at most 3.59, which is the currently best known approximation factor for ML.

Let $G = (\mathcal{D} \cup \{r\}, E)$ be the complete graph on $N = |\mathcal{D}| + 1$ nodes with edge weights $\{d_e\}$ that form a metric. Let r be the root node at which the path visiting the nodes must originate. We use e to index E and j to index the nodes. In both LPs, we have variables $x_{j,t}$ for $t \geq d_{jr}$ to denote if j is visited at time t (where t ranges from 1 to T); for convenience, we think of $x_{j,t}$ as being defined for all t , with $x_{j,t} = 0$ if $d_{jr} > t$. (As in Section 2.1, one can move to a polynomial-size LP losing a $(1 + \epsilon)$ -factor.)

A compact LP. As before, we use a variable $z_{e,t}$ to denote if e has been traversed by time t .

$$\begin{aligned} \min \quad & \sum_{j,t} tx_{j,t} \quad \text{subject to} \quad (LP1) \\ & \sum_t x_{j,t} \geq 1 \quad \forall j; \quad \sum_e d_e z_{e,t} \leq t \quad \forall t; \quad \sum_{e \in \delta(S)} z_{e,t} \geq \sum_{t' \leq t} x_{j,t'} \quad \forall t, S \subseteq \mathcal{D}, j; \quad x, z \geq 0. \end{aligned}$$

Theorem 3.1 *The integrality gap of (LP1) is at most 10.78.*

Proof : Let (x, z) be an optimal solution to (LP1), and $L_j^* = \sum_t tx_{j,t}$. For $\alpha \in [0, 1]$, define the α -point of j , $\tau_j(\alpha)$, to be the smallest t such that $\sum_{t' \leq t} x_{j,t'} \geq \alpha$. Let $D_t(\alpha) = \{j : \tau_j(\alpha) \leq t\}$. We round (x, z) as follows. We pick $\alpha \in (0, 1]$ according to the density function $q(x) = 2x$. At each time t , we utilize the $\frac{3}{2}$ -integrality-gap of the subtour-elimination LP for TSP and the parsimonious property (see [33, 30, 18, 6]), to round $\frac{2z}{\alpha}$ and obtain a tour on $\{r\} \cup D_t(\alpha)$ of cost $C_t(\alpha) \leq \frac{3}{\alpha} \cdot \sum_e d_e z_{e,t} \leq \frac{3t}{\alpha}$. We now use Lemma 3.2 to combine these tours.

Lemma 3.2 ([19] paraphrased) *Let $\text{Tour}_1, \dots, \text{Tour}_k$ be tours containing r , with Tour_i having cost C_i and containing N_i nodes, where $N_0 := 1 \leq N_1 \leq \dots \leq N_k = N$. One can find a subset $\text{Tour}_{i_1}, \dots, \text{Tour}_{i_b=k}$ of tours, and a way of concatenating them that gives total latency at most $\frac{3.59}{2} \sum_i C_i(N_i - N_{i-1})$.*

The tours we obtain for the different times are nested (as the $D_t(\alpha)$ s are nested). So $\sum_{t \geq 1} C_t(\alpha)(|D_t(\alpha)| - |D_{t-1}(\alpha)|) = \sum_j \sum_{t: j \in D_t(\alpha) \setminus D_{t-1}(\alpha)} C_t(\alpha) = \sum_j C_{\tau_j(\alpha)}(\alpha) \leq 3 \sum_j \frac{\tau_j(\alpha)}{\alpha}$. Thus, using Lemma 3.2, and taking expectation over α (note that $\mathbb{E}[\frac{\tau_j(\alpha)}{\alpha}] \leq 2L_j^*$), we obtain that the total latency-cost is at most $(3.59 \cdot 3) \sum_j L_j^*$. ■

Note that in the above proof we did not need any procedure to solve k -MST or its variants, but rather just needed the integrality gap for the subtour-elimination LP to be a constant. Also, we can modify the rounding procedure to ensure that (latency-cost of j) $\leq 18\tau_j(0.5)$ for each client j , as follows. (Such a guarantee is useful to bound the total cost when its measured as an \mathcal{L}_p norm for $p > 1$; see Section 4.) We now only consider times $t_\ell = 2^\ell$. Recall that for any $\alpha \in (0, 1)$, at each time t , we can obtain a tour on $\{r\} \cup D_t(\alpha)$ of cost $C_t(\alpha) \leq \frac{3t}{\alpha}$. We take this tour for t_ℓ , and traverse the resulting tour randomly clockwise or anticlockwise (this choice can easily be derandomized), and concatenate all these tours. Let

$\ell_j(\alpha)$ be the smallest ℓ such that $t_\ell \geq \tau_j(\alpha)$. So the (expected) latency-cost of j is at most $\sum_{\ell < \ell_j(\alpha)} \frac{3t_\ell}{\alpha} + \frac{1}{2} \cdot \frac{3t_{\ell_j(\alpha)}}{\alpha} \leq \frac{4.5t_{\ell_j(\alpha)}}{\alpha} \leq \frac{9\tau_j(\alpha)}{\alpha}$. Fixing $\alpha = 0.5$, we obtain a 36-approximation with the per-client guarantee that (latency-cost of j) $\leq 18\tau_j(0.5)$ for each j .

An exponential-size LP: relating the orienteering and latency problems. Let \mathcal{P}_t and \mathcal{T}_t denote respectively the collection of all (simple) paths and trees rooted at r of length at most t . For each path $P \in \mathcal{P}_t$, we introduce a variable $z_{P,t}$ that indicates if P is the path used to visit the nodes with latency-cost at most t .

$$\begin{array}{ll|ll}
\min & \sum_{j,t} tx_{j,t} & (\text{LP2}_{\mathcal{P}}) & \max & \sum_j \alpha_j - \sum_t \beta_t & (\text{LD2}) \\
\text{s.t.} & \sum_t x_{j,t} \geq 1 & \forall j & \text{s.t.} & \alpha_j \leq t + \sum_{t' \geq t} \theta_{j,t'} & \forall j, t \quad (7) \\
& \sum_{P \in \mathcal{P}_t} z_{P,t} \leq 1 & \forall t \quad (5) & & \sum_{j \in P} \theta_{j,t} \leq \beta_t & \forall t, P \in \mathcal{P}_t \quad (8) \\
& \sum_{P \in \mathcal{P}_t: j \in P} z_{P,t} \geq \sum_{t' \leq t} x_{j,t'} & \forall j, t \quad (6) & & \alpha, \beta, \theta \geq 0. & (9) \\
& x, z \geq 0. & & & &
\end{array}$$

(5) and (6) encode that at most one path may be chosen for any time t , and that every node j visited at time $t' \leq t$ must lie on this path. (LD2) is the dual LP with exponentially many constraints. Let (LP2 $_{\mathcal{T}}$) be the analogue of (LP2 $_{\mathcal{P}}$) with tree variables, where we have variables $z_{Q,t}$ for every $Q \in \mathcal{T}_t$, and we replace all occurrences of $z_{P,t}$ in (LP2 $_{\mathcal{P}}$) with $z_{Q,t}$.

Separating over the constraints (8) involves solving a (rooted) path-orienteering problem: for every t , given rewards $\{\theta_{j,t}\}$, we want to determine if there is a path P rooted at r of length at most t that gathers reward more than β_t . A (ρ, γ) -{path, tree} approximation algorithm for the path-orienteering problem is an algorithm that always returns a {path, tree} rooted at r of length at most $\gamma(\text{length bound})$ that gathers reward at least (optimum reward)/ ρ . Chekuri et al. [11] give a $(2+\epsilon, 1)$ -path approximation algorithm, whereas [10] design a $(1+\epsilon, 1+\epsilon)$ -tree approximation for orienteering (note that weighted orienteering can be reduced to unweighted orienteering with a $(1+\epsilon)$ -factor loss). We prove that even a (ρ, γ) -tree approximation algorithm for orienteering can be used to obtain an $O(\rho\gamma)$ -approximation for ML. First, we show how to compute a near-optimal LP-solution. Typically, one argues that, scaling the solution computed by the ellipsoid method run on the dual with the approximate separation oracle yields a feasible and near-optimal dual solution, and this is then used to obtain a near-optimal primal solution (see, e.g., [24]). However, in our case, we have negative terms in the dual objective function, which makes our task trickier: if our (unicriteria) ρ -approximate separation oracle determines that (α, β, θ) is feasible, then although $(\alpha, \rho\beta, \theta)$ is feasible to (LD2), one has *no guarantee on the value of this dual solution*. Instead, the notion of approximation we obtain for the primal solution computed involves bounded violation of the constraints.

Let (LP2 $_{\mathcal{P}}^{(a,b)}$) be (LP2 $_{\mathcal{P}}$) where we replace \mathcal{P}_t by \mathcal{P}_{bt} , and the RHS of (5) is now a . Let (LP2 $_{\mathcal{T}}^{(a,b)}$) be defined analogously. Let $OPT_{\mathcal{P}}$ be the optimal value of (LP2 $_{\mathcal{P}}$) (i.e., (LP2 $_{\mathcal{P}}^{(1,1)}$)). Note that $OPT_{\mathcal{P}}$ is a lower bound on the optimum latency.

Lemma 3.3 *Given a (ρ, γ) -tree approximation for the orienteering problem, one can compute a feasible solution (x, z) to (LP2 $_{\mathcal{T}}^{(\rho, \gamma)}$) of cost at most OPT .*

Proof : Lemma 3.3 Define

$$\begin{aligned} \text{Pfeas}(\nu; a, b) &:= \left\{ (\alpha, \beta, \theta) : (7), (9), \sum_{j \in P} \theta_{j,t} \leq \beta_t \quad \forall P \in \mathcal{P}_{bt}, \sum_j \alpha_j - a \sum_t \beta_t \geq \nu \right\}, \\ \text{Tfeas}(\nu; a, b) &:= \left\{ (\alpha, \beta, \theta) : (7), (9), \sum_{j \in Q} \theta_{j,t} \leq \beta_t \quad \forall Q \in \mathcal{T}_{bt}, \sum_j \alpha_j - a \sum_t \beta_t \geq \nu \right\}. \end{aligned}$$

Note that $OPT_{\mathcal{P}}$ is the largest value of ν such that $\text{Pfeas}(\nu; 1, 1)$ is feasible. Given $\nu, (\alpha, \beta, \theta)$, if there was an algorithm to either show $(\alpha, \beta, \theta) \in \text{Pfeas}(\nu; 1, 1)$ or exhibit a separating hyperplane, then using the ellipsoid method we could optimally solve for $OPT_{\mathcal{P}}$. However, such a separation oracle would solve orienteering exactly.

We use the (ρ, γ) -tree approximation algorithm to give an *approximate* separation oracle in the following sense. Given $\nu, (\alpha, \beta, \theta)$, we either show $(\alpha, \rho\beta, \theta) \in \text{Pfeas}(\nu; 1, 1)$, or we exhibit a hyperplane separating (α, β, θ) and $\text{Tfeas}(\nu; \rho, \gamma)$. Note that $\text{Tfeas}(\nu; \rho, \gamma) \subseteq \text{Pfeas}(\nu; 1, 1)$. Thus, for a fixed ν , the ellipsoid method in polynomial time, either certifies that $\text{Tfeas}(\nu; \rho, \gamma)$ is empty or returns a point (α, β, θ) with $(\alpha, \rho\beta, \theta) \in \text{Pfeas}(\nu; 1, 1)$. Let us describe the approximate separation oracle first, and then use the above fact to prove the lemma. First, check if $(\sum_j \alpha_j - \rho\beta_t \geq \nu)$, (7), and (9) hold, and if not, we use the appropriate inequality as the separating hyperplane between (α, β, θ) and $\text{Tfeas}(\nu; \rho, \gamma)$. Next, for each t , we run the (ρ, γ) -tree approximation on the orienteering problem specified by $(G, \{d_e\})$, root r , rewards $\{\theta_{j,t}\}$, and budget t . If for some t , we obtain a tree $Q \in \mathcal{T}_{\gamma t}$ with reward greater than β_t , then we return $\sum_{j \in Q} \theta_{j,t} \leq \beta_t$ as the separating hyperplane. If not, then for all paths P of length at most t in G , we have $\sum_{j \in P} \theta_{j,t} \leq \rho\beta_t$ and thus $(\alpha, \rho\beta, \theta) \in \text{Pfeas}(\nu; 1, 1)$.

We find the largest ν^* (via binary search) such that the ellipsoid method run for ν^* with our separation oracle returns a solution $(\alpha^*, \beta^*, \theta^*)$ with $(\alpha^*, \rho\beta^*, \theta^*) \in \text{Pfeas}(\nu^*; 1, 1)$; hence, we have $\nu^* \leq OPT_{\mathcal{P}}$ (by duality). Now for $\epsilon > 0$, the ellipsoid method run for $\nu^* + \epsilon$ terminates in polynomial time certifying the infeasibility of $\text{Tfeas}(\nu^* + \epsilon; \rho, \gamma)$. That is, it generates a polynomial number of inequalities of the form (7), (9), and $(\sum_{j \in Q} \theta_{j,t} \leq \beta_t)$ where $Q \in \mathcal{T}_{\gamma t}$, which together with the inequality $\sum_j \alpha_j - \rho \sum_t \beta_t \geq \nu^* + \epsilon$ constitute an infeasible system. Applying Farkas' lemma, equivalently, we get a polynomial sized solution (x, z) to $(\text{LP2}_{\mathcal{T}}^{(\rho, \gamma)})$ that has cost at most $\nu^* + \epsilon$. Taking ϵ small enough (something like $1/\exp(\text{input size})$) so that $\ln(1/\epsilon)$ is still polynomially bounded, this also implies that (x, z) has cost at most $\nu^* \leq OPT_{\mathcal{P}}$. This completes the proof of the lemma. ■

Theorem 3.4 (i) A feasible solution (x, z) to $(\text{LP2}_{\mathcal{T}}^{(\rho, \gamma)})$ (or the corresponding LP-relaxation for MGL) can be rounded to obtain a solution of expected cost at most $O(\rho\gamma) \cdot \sum_{j,t} tx_{j,t}$; (ii) A feasible solution (x, z) to $(\text{LP2}_{\mathcal{P}}^{(\rho, \gamma)})$ can be rounded to obtain a solution of cost at most $(3.59 \cdot 2)\rho\gamma \sum_{j,t} tx_{j,t}$.

Proof : We prove part (i) first. (Note that for ML, the analysis leading to Theorem 3.1 already implies that one can obtain a solution of cost at most $10.78\rho\gamma \sum_{j,t} tx_{j,t}$, because setting $\mathfrak{z}_{e,t} = \sum_{Q \in \mathcal{T}_{\gamma t}: e \in Q} z_{Q,t}$ yields a solution (x, \mathfrak{z}) that satisfies $\sum_e d_e \mathfrak{z}_{e,t} \leq \rho\gamma t$ and all other constraints of (LP1).) We sketch a (randomized) rounding procedure that also works for MGL and yields improved guarantees. We may assume that each $z_{Q,t} \in [0, 1]$. At each time $t_\ell = 2^\ell$, $\ell = \lceil \log_2 T + 4 \log_2 m \rceil$, we select at most $\lceil w_{t_\ell} \rceil$ trees from \mathcal{T}_{bt_ℓ} , picking each $Q \in \mathcal{T}_{\gamma t_\ell}$ with probability z_{Q,t_ℓ} . (We can always do this (efficiently) since for every time t , the polytope $\{\mathfrak{z} \in [0, 1]^{\mathcal{T}_{bt}} : \sum_Q \mathfrak{z}_{Q,t} \leq \lceil \sum_Q z_{Q,t} \rceil\}$ is integral.) We take the union of all these trees. Note that expected cost of the resulting subgraph is at most $w_t \gamma t_\ell \leq \rho\gamma t_\ell$. We “Eulerify” the resulting subgraph to obtain a tour for t_ℓ of cost at most $2\rho\gamma t_\ell$, and concatenate these tours. The probability that j is not visited (or covered) by the tour for t_ℓ is at most $1 - \sum_{t \leq t_\ell} x_{j,t}$, which implies that with high

probability, we obtain a tour spanning all nodes. Letting $\tau_j = t_j(\frac{2}{3})$, the expected latency-cost of j is at most $2\rho\gamma(2\tau_j + 2\tau_j \sum_{k \geq 0} (\frac{2}{3})^k) \leq 16\rho\gamma\tau_j$.

To prove part (ii), we adopt a rounding procedure that again utilizes Lemma 3.2, and yields the stated bound (deterministically). Recall that $\tau_j(\alpha)$ denotes the α -point of j . Let $D_t(\alpha) = \{j : \tau_j(\alpha) \leq t\}$. For any $\alpha \in (0, 1)$, and time t , we now show to obtain a tour spanning $D_t(\alpha) \cup \{r\}$ of cost at most $\frac{2\rho\gamma t}{\alpha}$. We can then proceed as in the rounding procedure for Theorem 3.1 to argue that, for the tours we obtain, the quantity $\sum_i C_i(N_i - N_{i-1})$ appearing in Lemma 3.2 is bounded by $2\rho\gamma \sum_j \frac{t_j(\alpha)}{\alpha}$. Hence, choosing α as before according to the distribution $q(x) = 2x$ and taking expectations, we obtain a solution with the stated cost.

Let K be such that $K\alpha$, $\{Kz_{P,t}\}_{P \in \mathcal{P}_{\gamma t}}$ are integers. For each P with $z_{P,t} > 0$, we create $Kz_{P,t}$ copies of each edge on P , and direct the edges away from r . Let $A_{P,t}$ denote the resulting arc-set. Note that in $A_t := \biguplus_{P: z_{P,t} > 0} A_{P,t}$, every node $j \in \mathcal{D}$ has in-degree at least its out-degree, and there are $K\alpha$ arc-disjoint paths from r to j for each $j \in D_t(\alpha)$. So applying Theorem 2.6 in Bang-Jensen et al. [4], one can obtain $K\alpha$ arc-disjoint out-arborescences rooted at r , each containing all nodes of $D_t(\alpha)$. Thus, if we pick the cheapest such arborescence and “Eulerify” it, we obtain a tour spanning $D_t(\alpha) \cup \{r\}$ of cost at most $2 \cdot K\rho\gamma t \cdot \frac{1}{K\alpha} = \frac{2\rho\gamma t}{\alpha}$. ■

In Appendix A, we prove an analogue of Lemma 3.3 for MGL. Combined with part (i) of Theorem LP2 \mathcal{P} , this shows that (even) a bicriteria approximation for “group orienteering” yields an approximation for MGL while losing a constant factor.

4 Extensions

We now consider various well-motivated extensions of MLUFL, and show that our LP-based techniques and algorithms are quite versatile and extend with minimal effort to yield approximation guarantees for these more general MLUFL problems. Our goal here is to emphasize the flexibility afforded by our LP-based techniques, and we have not attempted to optimize the approximation factors.

Monotone latency-cost functions with bounded growth, and higher \mathcal{L}_p norms. Consider the generalization of MLUFL, where we have a non-decreasing function $\lambda(\cdot)$ and the latency-cost of client j is given by $\lambda(\text{time taken to reach the facility serving } j)$; the goal, as before, is to minimize the sum of the facility-opening, client-connection, and client-latency costs. Say that λ has growth at most p if $\lambda(cx) \leq c^p \lambda(x)$ for all $x \geq 0$, $c \geq 1$. It is not hard to see that for concave λ , we obtain the same performance guarantees as those obtained in Section 2 (for $\lambda(x) = x$). So we focus on the case when λ is convex, and obtain an $O(\max\{(p \log^2 n)^p, p \log n \log m\})$ -approximation algorithm for convex latency functions of growth p . As a corollary, we obtain an $O(p \log n \max\{\log n, \log m\})$ -approximation for \mathcal{L}_p -MLUFL, where we seek to minimize the facility-opening cost + client-connection cost + the \mathcal{L}_p -norm of client-latencies.

Theorem 4.1 *There is an $O(\max\{(p \log^2 n)^p, p \log n \log m\})$ -approximation algorithm for MLUFL with convex monotonic latency functions of growth (at most) p .*

Proof : We highlight the changes to the algorithm and analysis in Section 2.1. We again assume that $T = \text{poly}(m)$ for convenience. This assumption can be dropped by proceeding as in Theorem 2.8; we do after proving the theorem. The objective of (P) now changes to $\min \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} (c_{ij} + \lambda(t)) x_{ij,t}$. The only change to Algorithm 1 is that we now define $t_\ell = \min\{2^{\ell/p}, T\}$ and set $\mathcal{N} := \lceil p \log_2(2^{1/p} \tau_{\max}) + 4 \log_2 m \rceil = O(p \log m)$. Define $\text{Lcost}_j = \sum_{i,t} \lambda(t) x_{ij,t}$. Note that we now have $\lambda(\tau_j) \leq 12 \text{Lcost}_j$. Define ℓ_j to be the first phase ℓ such that $t_\ell \geq \tau_j$. Let the random variable P_j be as defined in Lemma 2.5. The failure probability of the algorithm is again at most $1/\text{poly}(m)$. The facility-cost incurred in $O(p \log n \log m) \sum_{i,t} f_i y_{i,t}$,

and the connection cost of client j is at most $4C_j^*$. We generalize Lemma 2.5 below to show that $E[L_j] = O((p \log^2 n)^p) \lambda(t_{\ell_j}) \leq O((2p \log^2 n)^p) \cdot \lambda(\tau_j)$, which yields the desired approximation. We have

$$L_j \leq \lambda\left(\sum_{\ell \leq P_j} d(\text{Tour}_\ell)\right) \leq \lambda\left(O(\log^2 n) \sum_{\ell \leq P_j} t_\ell\right) \leq O(\log^2 n) \lambda\left(\sum_{\ell \leq P_j} t_\ell\right),$$

so $E[L_j] \leq O(\log^2 n) \left[\lambda\left(\sum_{\ell \leq \ell_j} t_\ell\right) + \sum_{k \geq 1} \Pr[P_j \geq \ell_j + k] \lambda\left(\sum_{\ell \leq \ell_j + k} t_\ell\right) \right]$. Now, $\Pr[P_j \geq \ell_j + k] \leq \left(\frac{4}{9}\right)^k$, $\sum_{\ell \leq \ell_j + k} t_\ell \leq t_{\ell_j} \cdot \frac{2^{k/p}}{1 - 2^{-1/p}}$, and $(2^{1/p} - 1) \geq \frac{\ln 2}{p}$. Plugging these in gives,

$$E[L_j] = O(\log^2 n) \cdot \frac{2}{(2^{1/p} - 1)^p} \cdot \lambda(t_{\ell_j}) \cdot \sum_{k \geq 0} \left(\frac{4}{9}\right)^k 2^k = O((p/\ln 2)^p \log^2 n) \lambda(t_{\ell_j}). \quad \blacksquare$$

Removing the assumption $T = \text{poly}(m)$ in Theorem 4.1. As in the case of Theorem 2.8, to drop the assumption that $T = \text{poly}(m)$, we (a) solve the LP considering only times in $\text{TS} = \{T_0, \dots, T_k\}$ (where $T_r = \lceil (1 + \epsilon)^r \rceil$); and (b) set $t_\ell = \text{TS}(\bar{L} \cdot 2^{\ell/p})$ and the number of phases to $\mathcal{N} := \lceil p \log_2(2^{1/p} \tau_{\max}/\bar{L}) + 4 \log_2 m \rceil$, where $\bar{L} = (\sum_{j,i,t} t x_{ij,t})/m$. Note that $\mathcal{N} = O(p \log m)$, and $\lambda(\bar{L}) \leq (\sum_{j,i,t} \lambda(t) x_{ij,t})/m = (\sum_j L \text{cost}_j)/m$, which shows that the expected latency cost incurred for clients j with $\tau_j \leq t_0$ is at most $\sum_j L \text{cost}_j$.

Corollary 4.2 *One can obtain an $O(p \log n \max\{\log n, \log m\})$ -approximation algorithm for \mathcal{L}_p -MLUFL.*

Proof : As is standard, we enumerate all possible values of the \mathcal{L}_p -norm of the optimal client latencies in powers of 2, losing potentially another factor of 2 in the approximation factor. To avoid getting into issues about estimating the solution-cost for a given guess (since our algorithms are randomized), we proceed as follows. For a given guess Lat , we solve (P) modifying the objective to be $\min \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} c_{ij} x_{ij,t}$, and we adding the constraint $\sum_{j,i,t} t^p x_{ij,t} \leq \text{Lat}^p$. Among all such guesses and corresponding optimal solutions, let (x, y, z) be the solution that minimizes $\sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} c_{ij} x_{ij,t} + \text{Lat}$. Let OPT denote this minimum value. Note that $OPT \leq 2O^*$, where O^* is the optimum value of the \mathcal{L}_p -MLUFL instance. We apply Theorem 4.1 (with $\lambda(x) = x^p$) to round (x, y, z) . Let F , C , and $L = \sum_j L_j$, denote respectively the (random) facility-opening, connection-, and latency- cost of the resulting solution. The bounds in Theorem 4.1 imply that $E[(\sum_j L_j)^{1/p}] \leq (E[\sum_j L_j])^{1/p} \leq O(p \log^2 n) \text{Lat}$, which combined with the bounds on $E[F]$ and $E[C]$, shows that the expected total cost is $O(p \log n \max\{\log n, \log m\}) OPT$. \blacksquare

We obtain significantly improved guarantees for related MLUFL, metric uniform MLUFL, and ML with (convex) latency functions of growth p . For related MLUFL and ML, the analyses in Sections 2.2 and 3 directly yield an $O(2^{O(p)})$ -approximation guarantee since for both problems, we can (deterministically) bound the delay of client j by $O(\alpha\text{-point of } j)$ (for suitable α). For metric uniform MLUFL, we obtain an $O(1)$ -approximation bound as a consequence of Theorem 2.12: this follows because one can devise an $O(1)$ -approximation algorithm for the zero-facility-cost version of the problem by adapting the ideas used in [3]. Thus, we obtain an $O(1)$ -approximation for the \mathcal{L}_p -versions of related-MLUFL, metric uniform MLUFL, and ML.

k -route MLUFL with length bounds. All our algorithms easily generalize to k -route length-bounded MLUFL, where we are given a budget B and we may use (at most) k paths starting at r of $(d-)$ length at most B to traverse the open facilities and activate them. This captures the scenario where one can use k vehicles in parallel, each with capacity B , starting at the root depot to activate the open facilities. Observe that with $B = \infty$, we obtain a generalization of the k -traveling repairmen problem considered in [14].

We modify (P) by setting $T = B$ and setting the RHS of (3) to kt . In Algorithm 1, we now obtain a tour Tour_ℓ in phase ℓ of (expected length) $O(\log^2 n)kt_\ell$. Each facility $i \in \text{Tour}_\ell$ satisfies $\sum_{t \leq t_\ell} y_{i,t} > 0$,

so $d(i, r) \leq t_\ell$, and we may therefore divide Tour_ℓ into k tours of length at most $O(\log^2 n)t_\ell$. Thus, we obtain the same guarantee on the expected cost incurred, and we violate the budget by an $O(\log^2 n)$ -factor, that is, we get a bicriteria $(\text{polylog}, O(\log^2 n))$ -approximation. Similarly, for related MLUFL and ML, we obtain an $(O(1), O(1))$ -approximation. For ML, we may again use either (LP1) or (LP2_P): in both LPs we set $T = B$; in (LP1), we now have the constraint $\sum_e d_e z_{e,t} \leq kt$, and in (LP2_P), the RHS of (5) is now k . For metric uniform MLUFL, we modify (Unif-P) in the obvious way: we now have $\sum_t y_{i,t} \leq k$ for each time t , and t now ranges from 1 to B . We can again apply Theorem 2.12 here to obtain a (*unicriteria*) $O(1)$ -approximation algorithm: for the zero-facility-location problem (where we may now “open” at most kB facilities), we can adapt the ideas in [3] to devise an $O(1)$ -approximation algorithm.

Finally, these guarantees extend to latency functions of bounded growth (in the same way that guarantees for MLUFL extend to the setting with latency functions). Thus, in particular, we obtain an $O(1)$ -approximation algorithm for the \mathcal{L}_p -norm k -traveling repairmen problem; this is the *first* approximation guarantee for this problem.

Non-uniform latency costs. We consider here the setting where each client j has a (possibly different) time-to-cost conversion factor λ_j , which measures j ’s sensitivity to time delay (vs. connection cost); so the latency cost of a client j is now given by $\lambda_j t_j$, where t_j is the delay faced by the facility serving j .

All our guarantees in Sections 2.2, 2.3, and 3 continue to hold in this non-uniform latency setting. In particular, we obtain a constant approximation guarantee for related metric MLUFL, metric uniform MLUFL, and ML. Notice that the metric uniform MLUFL problem cannot now be solved via a reduction to the metric-UFL variant discussed in Section 2.3; however we can still use Corollary 2.13 to obtain a 10.773-approximation. For general MLUFL, it is not hard to see that our analysis goes through under the assumption $T = \text{poly}(m)$. However, the scaling trick used to bypass this assumption leads to an extra $O(\log(\frac{\lambda_{\max}}{\lambda_{\min}}))$ factor in the approximation.

Recall that the scaling factor \bar{L} (in the definition of t_ℓ) in Section 2.1 was defined as $\sum_j L_j^*/m$, where $L_j^* = \sum_{i,t} tx_{ij,t}$. Now, $L_j^* = \sum_{i,t} \lambda_j tx_{ij,t}$, and we set $\bar{L} = \sum_j L_j^* / \sum_j \lambda_j$. One can again argue that the expected latency-cost of each client j is at most $\lambda_j \cdot O(\log^2 n) \cdot \max\{\bar{L}, \tau_j\}$, so we incur an $O(\log^2 n)$ -factor in the latency-cost. The number of phases, however, is $\mathcal{N} := \lceil \log_2(2\tau_{\max}/\bar{L}) + 4\log_2 m \rceil$, and $2\tau_{\max}/\bar{L} = O(\frac{\tau_{\max} \sum_j \lambda_j}{\sum_j L_j^*}) \leq O(\frac{m\lambda_{\max}}{\lambda_{\min}})$, which gives an extra $\log_2(\lambda_{\max}/\lambda_{\min})$ factor in the facility-opening cost.

Finally, as before, these guarantees also translate to the k -route length-bounded versions of our problems.

References

- [1] A. Archer, A. Levin, and D. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM J. Comput.*, 37(5):1472–1498, 2008.
- [2] Y. Azar, I. Gamzu, and X. Yin. Multiple intents re-ranking. In *Proceedings of 41st STOC*, pages 669–678, 2009.
- [3] I. Bae, R. Rajaraman, and C. Swamy. Approximation algorithms for data placement problems. *SIAM Journal on Computing*, 38(4):1411–1429, 2008.
- [4] J. Bang-Jensen, A. Frank, and B. Jackson. Preserving and increasing local edge-connectivity in mixed graphs. *SIAM Journal on Discrete Mathematics*, 8(2):155–178, 1995.
- [5] N. Bansal, A. Gupta, and R. Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *Proceedings of 21st SODA*, pages 1539–1545, 2010.

- [6] D. Bienstock, M. Goemans, D. Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. *Math. Program.*, 59:413–420, 1993.
- [7] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The Minimum Latency Problem. In *Proceedings of 26th STOC*, pages 163–171, 1994.
- [8] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Proceedings of the 10th APPROX*, pages 29–43, 2007.
- [9] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for Group Steiner Trees and k-median. In *Proceedings of the 30th STOC*, pages 114–123, 1998.
- [10] K. Chaudhuri, P. B. Godfrey, S. Rao, and K. Talwar. Paths, Trees and Minimum Latency Tours. In *Proceedings of 44th FOCS*, pages 36–45, 2003.
- [11] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. In *Proceedings of 19th SODA*, pages 661–670, 2008.
- [12] F. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.
- [13] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *Proceedings of 19th SODA*, pages 1174–1183, 2008.
- [14] J. Fakcharoenphol, C. Harrelson, and S. Rao. The k -travelling repairman problem. *ACM Trans. on Alg.*, Vol 3, Issue 4, Article 40, 2007.
- [15] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of 35th STOC*, pages 448–455, 2003.
- [16] U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.
- [17] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.
- [18] M. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60:145–166, 1993.
- [19] M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. In *Proceedings of 7th SODA*, pages 152–158, 1996.
- [20] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flow. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 389–398, 2001.
- [21] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. *Approximation Algorithms for Optimal Decision Trees and Adaptive TSP Problems*. arXiv:1003.0722, to appear in ICALP 2010.
- [22] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of 35th STOC*, pages 585–594, 2003.

- [23] J. H. Lin and J. S. Vitter. ϵ -approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 771–782, 1992.
- [24] K. Jain, M. Mahdian, and M. Salavatipour. Packing Steiner trees. In *Proceedings of 14th SODA*, pages 266–274, 2003.
- [25] P. Mirchandani and R. Francis, eds. *Discrete Location Theory*. John Wiley and Sons, Inc., New York, 1990.
- [26] V. Nagarajan. *Approximation Algorithms for Sequencing Problems*. Ph.D. thesis, Tepper School of Business, Carnegie Mellon University, 2009.
- [27] R. Ravi and F. S. Selman. Approximation algorithms for the traveling purchaser problem and its variants in network design. In *Proceedings of the 7th Annual European Symposium on Algorithms*, pages 29–40, 1999.
- [28] D. Shmoys, R. Levi, and C. Swamy. Facility location with service installation costs. In *Proceedings of 15th SODA*, pages 1081–1090, 2004.
- [29] D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 265–274, 1997.
- [30] D. Shmoys and D. Williamson. Analyzing the Held-Karp TSP bound: a monotonicity property with application. *Inf. Process. Lett.*, 35(6):281–285, 1990.
- [31] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.
- [32] P. Toth and D. Vigo, eds. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
- [33] L. Wolsey. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Study*, 13:121–134, 1980.

A Proofs omitted from the main body

Proof of Theorem 2.9 : Consider a GST instance $(H = (V, E), r, \{d_e\}_{e \in E}, \{G_j \subseteq V\}_{j=1}^m)$. Let $n = |V|$. We may assume that H is the complete graph, d is a metric, and the groups are disjoint. We abbreviate $\rho_{n,m}$ to ρ below. Let \mathcal{Q} denote the collection of all solutions to the path-variant of GST; that is, \mathcal{Q} consists of all paths starting at r that visit at least one node of each group. For a path $Q \in \mathcal{Q}$ and a group G_j , define Q_j to be the portion of Q from r to the first node of G_j lying on Q . Let $d_j(Q) = \sum_{e \in Q_j} d_e$ be the length of Q_j ; that is, $d_j(Q)$ is the latency of group j along path Q . Consider the following LP-relaxation for the path-variant of GST, and its dual. We have a variable x_Q for every $Q \in \mathcal{Q}$ indicating if path Q is chosen. We use Q below to index the paths in \mathcal{Q} .

$$\begin{array}{ll}
 \min & M \\
 \text{s.t.} & \sum_Q d_j(Q) x_Q \leq M \quad \forall j \\
 & \sum_Q x_Q \geq 1 \\
 & x \geq 0.
 \end{array} \quad (\text{P}') \quad \left| \quad \begin{array}{ll}
 \max & \alpha \\
 \text{s.t.} & \sum_j \lambda_j d_j(Q) \geq \alpha \quad \forall Q \\
 & \sum_j \lambda_j \leq 1 \\
 & \alpha, \lambda \geq 0.
 \end{array} \quad (\text{D}') \quad (10)$$

(P') has an exponential number of variables. But observe that separating over the constraints (10) in the dual involves solving an MGL problem. The minimum value (over all $Q \in \mathcal{Q}$) of the LHS of (10) is the optimal value of the MGL problem defined by $\{(G_j, \lambda_j)\}$, where we seek to minimize the weighted sum of client latency costs. (This weighted group latency problem can be reduced to the unweighted problem by “creating” λ_j copies of each group G_j (we can scale the λ_j s so that they are integral); equivalently (instead of explicitly creating copies), one can simulate this copying-process in whatever algorithm one uses for (unweighted) MGL.) Thus, a ρ -approximation algorithm for MGL yields a ρ -approximate separation oracle for (D'). Now, applying an argument similar to the one used by Jain et al. [24] shows that one can use this to (also) obtain a ρ -approximate solution (x, M) to (P').

We now use randomized rounding to round (x, M) and obtain a group Steiner tree of cost at most $O(\log m)M$. We pick path Q independently with probability $\min\{4 \log m \cdot x_Q, 1\}$. Let $\mathcal{Q}' \subseteq \mathcal{Q}$ denote the collection of paths picked. Note that for every group G_j , we have $\sum_{Q \in \mathcal{Q}: d_j(Q) \leq 2M} x_Q \geq \frac{1}{2}$. So a standard set-cover argument shows that with probability at least $1 - 1/m$, for every j , there is some path $Q^{(j)} \in \mathcal{Q}'$ such that $d_j(Q^{(j)}) \leq 2M$. We may assume that $\sum_Q x_Q = 1$, so Chernoff bounds show that $|\mathcal{Q}'| = O(\log m)$ with overwhelming probability. The group Steiner tree \mathcal{T} consists of the union of all the $Q_j^{(j)}$ (sub)paths (deleting edges to remove cycles as necessary). Clearly, the cost of \mathcal{T} is at most $|\mathcal{Q}'| \cdot 2M = O(\log m)M$. Note that \mathcal{T} also yields a path of length $O(\log m)M$ starting at r and visiting all groups, so the integrality gap of (P') is $O(\log m)$. ■

Extension of Lemma 3.3 to MGL. Notice that we did not use anything specific to the minimum-latency problem in the proof, and so essentially the same proof also applies to MGL. Recall that in MGL, we have a set \mathcal{F} of facilities and a d -metric on $\mathcal{F} \cup \{r\}$, and a collection of m groups $\{G_j \subseteq \mathcal{F}\}$. Analogous to $(\text{LP2}_{\mathcal{P}}^{(a,b)})$, the LP-relaxation with path variables for MGL and its dual are as follows.

$$\begin{array}{ll|ll}
\min & \sum_{j,t} tx_{j,t} & (\text{LP}'_{\mathcal{P}}^{(a,b)}) & \max & \sum_j \alpha_j - a \sum_t \beta_t & (\text{LD}'_{\mathcal{P}}(a,b)) \\
\text{s.t.} & \sum_t x_{j,t} \geq 1 & \forall j & \text{s.t.} & \alpha_j \leq t + \sum_{t' \geq t} \theta_{j,t'} & \forall j, t \\
& \sum_{P \in \mathcal{P}_{bt}} z_{P,t} \leq a & \forall t & & \sum_{j: G_j \cap P \neq \emptyset} \theta_{j,t} \leq \beta_t & \forall t, P \in \mathcal{P}_{bt} \quad (11) \\
& \sum_{P \in \mathcal{P}_{bt}: G_j \cap P \neq \emptyset} z_{P,t} \geq \sum_{t' \leq t} x_{j,t'} & \forall j, t & & \alpha, \beta, \theta \geq 0. & \\
& x, z \geq 0. & & & &
\end{array}$$

The LP-relaxation $(\text{LP}'_{\mathcal{T}}^{(a,b)})$ with tree variables is obtained by replacing \mathcal{P}_{bt} with \mathcal{T}_{bt} in $(\text{LP}'_{\mathcal{P}}^{(a,b)})$. The orienteering problem that we need to solve now to separate over the constraints (11) is *group orienteering*: given a reward $\theta_{j,t}$ for each group G_j , we want to determine if there is a path (or tree) rooted at r of length at most bt such that the total reward of the groups covered by it is more than β_t . Given these changes, the proof that one can obtain a feasible solution (x, y) to $(\text{LP}'_{\mathcal{T}}^{(a,b)})$ of cost at most the optimal value of $(\text{LP}'_{\mathcal{P}}^{(1,1)})$ is as in the proof of Lemma 3.3.